NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGNING A RELATIONAL DATABASE TO FACILITATE
MILITARY FAMILY HOUSING UTILIZATION DECISIONS

by

Jonalan Brickey

June 2001

Thesis Advisor: Daniel R. Dolk
Associate Advisor: James E. Illingworth

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 2001	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) Designing A Relational Database To Facilitate Military Family Housing Utilization Decisions		5. FUNDING NUMBERS	
6. AUTHOR(S) Brickey, Jonalan			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Army TRADOC Analysis Center Monterey, CA 93943		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

   The Housing Welcome Center (HWC) in Monterey, CA manages more than 2,200 military family houses and provides services to approximately 4,500 service members. The current housing information system consists of a proprietary legacy database system, a homemade non-relational database, a collection of word processing documents, and a static online application Web site. This semi-manual system can introduce errors, inconsistencies, and redundancies that may lead to frustration and possibly a decrease in quality of life.

   This research presents a prototype Housing Assignments and Terminations System (HATS) that uses a top-down systems analysis design approach. The research defines requirements and models that are transformed into a relational database with connectivity to a Web site. A graphical user interface (GUI) provides intuitive menus to manipulate and view data. Decision support tools help decision-makers visualize the data in a spatial manner and make better decisions to improve the quality of life for service members.

14. SUBJECT TERMS Database, Decision Support Tools, Legacy System, GIS, Military Family Housing			15. NUMBER OF PAGES 164
			16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

i

THIS PAGE INTENTIONALLY LEFT BLANK

DESIGNING A RELATIONAL DATABASE TO FACILITATE MILITARY FAMILY HOUSING UTILIZATION DECISIONS

Jonalan Brickey
Captain, United States Army
B.S., United States Military Academy, 1991

Submitted in partial fulfillment of the
requirements for the degree of

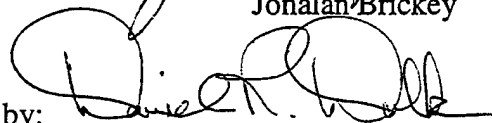## MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the
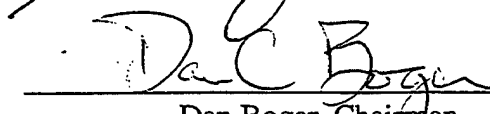
NAVAL POSTGRADUATE SCHOOL
June 2001

Author: _____
Jonalan Brickey

Approved by: _____
Daniel R. Dolk, Thesis Advisor

_____
James E. Illingworth, Associate Advisor

_____
Dan Boger, Chairman
Information Systems Academic Group

iii

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Housing Welcome Center (HWC) in Monterey, CA manages more than 2,200 military family houses and provides services to approximately 4,500 service members. The current housing information system consists of a proprietary legacy database system, a homemade non-relational database, a collection of word processing documents, and a static online application Web site. This semi-manual system can introduce errors, inconsistencies, and redundancies that may lead to frustration and possibly a decrease in quality of life.

This research presents a prototype Housing Assignments and Terminations System (HATS) that uses a top-down systems analysis design approach. The research defines requirements and models that are transformed into a relational database with connectivity to a Web site. A graphical user interface (GUI) provides intuitive menus to manipulate and view data. Decision support tools help decision-makers visualize the data in a spatial manner and make better decisions to improve the quality of life for service members.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

AEM	Applied Engineering Management
ASP	Active Server Pages
DD Form 1746	Application For Assignment To Housing
DLI	Defense Language Institute
DoD	Department of Defense
DON	Department of the Navy
DSS	Decision Support System
DST	Decision Support Tool
EFH	Electronic Family Housing
ESRI	Environmental Systems Research Institute Inc.
FAMIS	Family Activity Management Information System
GIS	Geographic Information System
HATS	Housing Assignments and Terminations System
HD	Housing Director
HDBA	Housing Database Administrator
HFS	Housing Facilities Supervisor
HMS	Housing Management Specialist
HWC	Housing Welcome Center
HWS	Housing Web Server
KAT	Knowledge Analysis of Tasks
LM	La Mesa (Housing Area)
MFH	Military Family Housing (or House)
NAVFACENGCOM	Naval Facilities Engineering Command
NPS	Naval Postgraduate School
NSA	Naval Support Activity
NSAMB	Naval Support Activity Monterey Bay
NSAMBINST 11101.4	NSA Instruction for Military Family Housing
ODBC	Open Database Connectivity
OMC	Ord Military Community
POM	Presidio of Monterey
QBE	Query by Example
Quarters	An individual house or apartment.
SM	Service Member (military or government)
SOM	Semantic Object Model
TRAC	Training and Doctrine Command Analysis Center
UML	Unified Modeling Language

I. INTRODUCTION

A. BACKGROUND

The Naval Support Activity Monterey Bay (NSAMB) manages the Department of Defense (DOD) Military Family Housing (MFH) Program in the Monterey, CA area. Within the NSAMB, the Assignments and Terminations branch operates the Housing Welcome Center (HWC), which is the interface between the organization and its customers—military and government service members (SMs).

The ability of the HWC to efficiently store and retrieve information about MFH and customers directly affects the quality of service provided to the customer, as well as the cost for the government to provide housing services. Many times, moving out of one home and finding another is the most stressful part of a family move. Every two or three years, most military and government service families pack up all of their belongings and move around the world to a new duty assignment. Sometimes, they live out of their suitcases in hotels, with their pets in a kennel, waiting anxiously for the day when they can move into a new home. If the HWC does its job correctly, SM families enjoy a quick transition to their new homes. If the HWC fails to do its job, the government wastes money providing excessive temporary lodging expenses while the stress levels for SM families grow larger by the day.

The HWC provides several services to SM families. The most routine service is assigning SM families to government-owned quarters, or MFH, and terminating the assignments when they depart. However, not every SM lives in MFH because it's not feasible for DOD to build and maintain quarters for everyone. The HWC also provides a

1

referral service to help some SMs find living accommodations in privately owned dwellings.

Counselors at the HWC currently use the Family of Management Information Systems (FAMIS) to manage the MFH Program. The requirements for the MFH Program have changed over the past few years, prompting the Housing System Administrator to find ways to modify FAMIS. Unfortunately, FAMIS is a proprietary system which requires cooperation from the highest levels of Navy Housing and the original FAMIS contractor to update the system. Since FAMIS is not a flexible application, the HWC has created three separate databases and added other applications, such as Borland's Report Smith application and Microsoft Office Suite, to augment FAMIS and provide MFH requirements.

As part of an ongoing effort to support military agencies in the local area, the Army's Training Analysis Center in Monterey, TRAC-Monterey, supports the HWC with forecasting tools and technical support. TRAC-Monterey recently evaluated the current MFH Assignments and Terminations Information System and proposed a new database system that supports web-based input from customers and provides more functionality.

B.   OBJECTIVES

The primary question to be addressed by this research is:

- What design methodology is appropriate for the web-based database design process?

Secondary questions to be answered by this research are:

- What data quality issues need to be addressed in a migration plan to move from the current file-based system to implementation of an improved database system?

- How can a relational database-driven system provide the Housing Welcome Center (HWC) with user-friendly views and reports, as well as flexibility and scalability to make facility management decisions effectively and efficiently?

- What decision support tool(s) can be used to augment the database application?

To highlight the benefits associated with modern database applications, we will create a relational database system with connectivity to the HWC Web site. Specifically, we will develop a prototype Housing Assignments and Terminations System (HATS) using Microsoft Access and Internet Information Server (IIS) to track assignments and other functions, available units, waiting service members, and housing utilization. In addition, we will employ decision support tools to help the housing office make better utilization decisions and improve customer service.

C.   SCOPE AND METHODOLOGY

This thesis focuses on designing a new assignments and terminations system for MFH in Monterey, California. We will provide a prototype database with all of the reports currently required by the HWC, using real data and web input from the customer. This thesis will not provide a full-scale implementation at the NSAMB HWC.

We will use the following systems analysis approach to develop the housing assignments and terminations relational database prototype:

- Requirements Analysis - This phase involves interviews with users and analysis of the current system to determine requirements. We will study business processes, review reports and forms, relevant business rules, and design use cases to show interactions between users and the system.

- Logical Database Design – Once the requirements are understood, we will use semantic object modeling to represent the relevant objects and their relationships. To complete this phase, we will convert the schema and object model into a full relational schema implemented in Microsoft Access.

3

- Physical Database Design - The physical database design requires converting the logical database design into normalized tables and relationships.

- Implementation and Testing - The database will be implemented in Microsoft Access. The Housing Web site will connect to the database using Microsoft's Open Database Connectivity standard (ODBC) and Internet Information Server (IIS).

- Decision Support Tools Analysis – Apply decision support tools to the database management system using Access and a Geographic Information System (GIS).

D.   ORGANIZATION

This thesis is organized as follows. Chapter II provides an overview of the current family housing assignments and terminations system at NSAMB. It discusses the original development and the known issues with the current system.

Chapter III proposes a reengineered system for housing assignments and terminations. It provides a detailed requirement analysis, using use cases, interviews, and process analysis.

Chapter IV details the database design and development process of the HATS prototype for the NSAMB.

Chapter V discusses data migration and data quality. It explains the challenges of cleaning data, transferring it to another system, and resolving data inconsistencies.

Chapter VI details the Graphical User Interface (GUI) design and development process. It also discusses Internet connectivity between the Web GUI and the prototype database.

Chapter VII discusses the potential utility of certain decision support tools in housing policies and procedures, and how such tools might be integrated with the HATS prototype.

Chapter VII provides a summary of conclusions and recommendations for further development.

THIS PAGE INTENTIONALLY LEFT BLANK

II. CURRENT HOUSING SYSTEM

A. INTRODUCTION

The purpose of an HWC assignments and terminations system is to assist the housing office in providing housing to eligible Department of Defense (DOD) employees. The system also serves as the automated database for maintaining records of military and privately owned housing facilities, eligible service members and dependents, and for providing forms and reports to facilitate record keeping. The current information management system used by the Housing Office is the Family Housing Activity Management Information System (FAMIS). Specifically, the assignments and terminations module is named "Families".

Unfortunately, the HWC has the Herculean task of knowing the real-time status of thousands of houses and service members. The HWC relies on FAMIS to record information about such things as the condition of MFH, the availability of privately owned property, the needs of the customer, and many other important factors. In addition, FAMIS has to interact with HWC counselors to provide timely and accurate information. The process of assignments and terminations inherently places large demands on FAMIS, which was not designed to keep pace with changing demands and evolving technology. Table 2-1 summarizes the HWC's customer and inventory statistics.

Service Members Supported	4,500
Assignments/Terminations (monthly)	80
Customer Referrals (monthly)	50
Number of MFH Units	2,200
Number of Referral Properties	200-500

Table 2-1.   Customers and Inventory Serviced by the HWC in Monterey, CA

B.   ASSIGNMENTS AND TERMINATIONS PROCESS

The process of assigning and terminating quarters is simple, yet complex, due to numerous DoD housing regulations.

In the assignment process, all MFHs are divided into categories, matching the rank structure of the military and government service members (SMs). Since there are normally more SMs than there are available houses, each category of MFH has a waiting list. The date that the SM departed his last duty assignment determines the SM's position on the waiting list. Whenever a MFH becomes available, the Housing Office assigns the next SM on the waiting list to the available MFH.

In the termination process, the SM files an "Intent to Vacate" report to housing as soon as he knows his moving date. The HMS schedules appointments for the HFS to inspect the MFH for cleanliness and suitability for the next occupant. Upon approval from the HFS, the SM obtains a termination letter from the HMS, making the MFH available for move-out maintenance (and eventually, available for occupancy).

This is a simplified explanation of the assignment and termination processes. Usually, there are many factors, such as medical conditions, priority duty positions, and other family needs that complicate the process and tax the information system.

C. FAMILY HOUSING ACTIVITY MANAGEMENT INFORMATION SYSTEM (FAMIS)

Originally developed by Applied Engineering Management (AEM) Corporation in 1985 for The Naval Facilities Engineering Command (NAVFACENGCOM), FAMIS was designed to manage as many Housing Office functions as possible. While the FAMIS suite has many modules for different housing functions, the "Families" module supports the assignments and terminations branch functions. The purpose of the "Families" module of FAMIS is to assist housing managers throughout the Navy in their efforts to supply military service members and their families both Navy-owned and private sector housing. For Navy-owned housing, it supports wait list management, assignment to and termination of quarters, occupancy, and utilization reporting and analysis. For private sector housing, it maintains up-to-date listings of available rental and sale properties, customized to the service member's criteria.

Over the last sixteen years, FAMIS has undergone many facelifts in an attempt to meet the growing needs of housing offices and utilize advances in information and computing technologies. The Navy admits that, as in many civilian and government software projects, FAMIS project management has been less than perfect. FAMIS was originally designed as a disk operating system (DOS) product. Due to poor program management, FAMIS only captured about half of the users' requirements. Around 1995, HQ Navy Housing decided not to upgrade FAMIS to a Windows application because the newer GUI was not proven; however, the contractor eventually made the database accessible through Microsoft Windows, using a DOS GUI emulation. At the same time,

the Navy decided to delay all major upgrades to FAMIS and build a new system using modern design methods (Leidle, 2001).

Since there were no planned upgrades to FAMIS, counselors at the Housing Office asked for, and finally received, other software to complement FAMIS to achieve their business needs. They received Borland's Report Smith to improve the quality and ease of generating reports. Later, they developed a Microsoft Access database because the FAMIS module could not provide availability reports.

FAMIS was originally designed as a total housing solution, meaning that it was intended to do everything for everyone in the housing domain. Unfortunately, as many ERP-like programs do, the application attempted to do too much, and in the process failed to do any individual module really well. Because the application attempts to cover all Housing Office functions, it contains over 85 separate database files to support the main application.

Module Name	Module Function
Maintenance	To manage housing maintenance: routine and emergency repairs, planned remodeling, and move-out inspections. This module also attempted to assist Facilities Managers with employee time cards, acquiring building supplies, and preparing bids. (The Navy is replacing this module with the Maximo application.)
Families	To assist housing in its efforts to supply SMs and family members either Navy owned or private sector housing.

This module is the primary focus of my research. |
Warehouse	To manage the issue and return of warehouse inventory items, such as furniture, tools, and outdoor equipment.
Inventory	To manage records for MFH and support facilities to be used by the Maintenance, Families, and Warehouse modules.
Administration	To customize the application for each installation.

Table 2-2.   FAMIS Module Names and Functions

D.   KNOWN ISSUES

Although there are many flaws with the current system, the following known issues are the most important to the users of the system.

1.   Poor User Interface

The FAMIS menus, designed using text-based interfaces over 15 years ago for the original DOS version, are outdated and difficult to navigate. In the Referral section, users cannot modify property information displayed on their monitors and they cannot just click on a modify button to navigate to the modify screen. Instead, they have to

11

just click on a modify button to navigate to the modify screen. Instead, they have to

navigate back to the main Referral menu and select the modify function. In addition, if a

user makes a mistake in the input/edit screen, s/he cannot correct the error without

closing the screen and navigating back to the same screen to modify the

record.
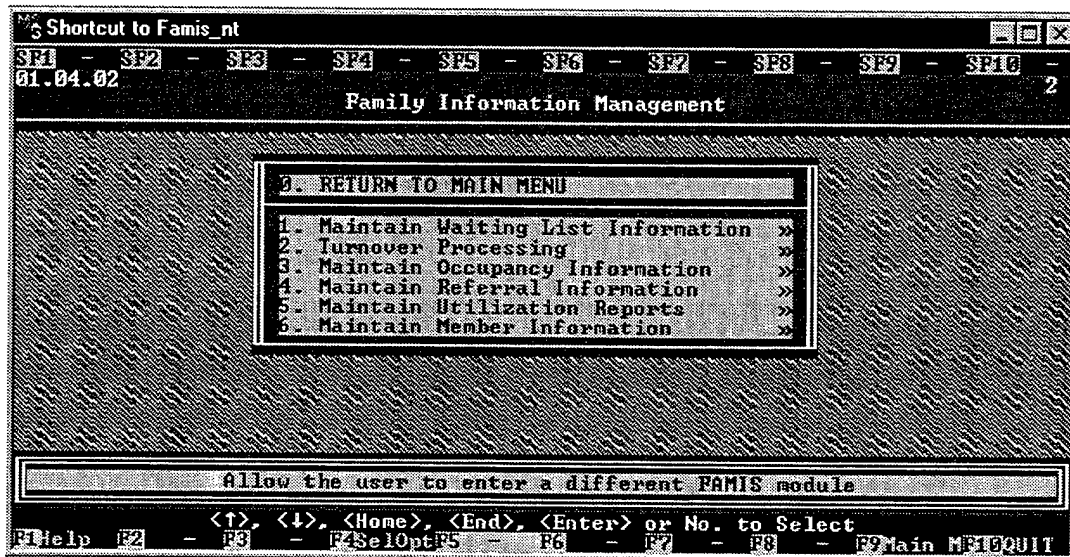
Figure 2.   FAMIS Main Menu and Graphical User Interface (GUI)

2.   No Connectivity with The Online Housing Application (DD Form 1746)

Since most SMs use the Internet daily at work and at home, the Housing Office

created an online housing application form to facilitate the application process.

Unfortunately, the current system cannot communicate with the Web server, so the

application information has to be sent to an HMS in text email messages. Once the HMS

receives the text message, s/he has to re-type all of the SM's information into FAMIS.

3.   Poor Asset Visibility Functions

Within several of the functional areas of FAMIS, users cannot complete actions

using FAMIS alone. They have to rely on patches and "work-arounds" to do their

12

Microsoft Access or Excel and they have to open Borland's Report Smith to conduct queries and reports. These patches cause redundancy and wasted time. For example, when I recently filed an "intent to vacate" to inform housing of my imminent move, the HMS clerk had to open three different applications (and three separate files) in order to tell the system that my quarters are being vacated, and to schedule inspection appointments. Here are the steps she performed:

- Open Excel and enter SM name and inspection dates.
- Open FAMIS and enter the same information.
- Open the Access "intent to vacate" file logbook and type the following: name, date, address, floor plan, phone #, inspection dates, and availability code.

Upon actually moving out of my government house, the HMS will have to open the Access vacate file, copy the information into Microsoft Word, and type a termination letter.

The Housing Office maintains two different Access logbooks, one to record when a SM accepts MFH—the "who's in the house" file, and one to record the "intent to vacate" information—the "who's leaving" file. In addition, there is no automated process for scheduling the inspection appointments.

4.   Poor Method of Assigning a SM with an MFH

FAMIS has no automated method of making a housing assignment. The HMSs have to print asset reports and wait list reports, then compare them manually to make housing assignments.

5.   Poor Reporting Functionality

FAMIS does not facilitate the creation of ad hoc reports. After implementing FAMIS, the Navy noticed that it was difficult to create queries and reports from the database. The Navy's solution was to augment FAMIS with Borland's Report Smith, a

13

commercial off-the-shelf (COTS) visual database reporting and query tool. It is not convenient for users to switch from FAMIS to Report Smith to view reports and query the database.

6. Poor Interoperability with Other Applications

The Department of the Navy Information Standards Guidance (IT-21) specifies the information management environment for all Navy commands. The NSAMB Housing Office uses the Microsoft Office in compliance with the IT-21 standard. However, FAMIS does not meet the IT-21 standard because it uses a text-based interface and outdated data exchange standards (which are not compatible with MS Office). Because FAMIS is not compatible with MS Office, the HMSs have to retype information from FAMIS into MS Office in order to type housing documents.

E. UPGRADING THE CURRENT SYSTEM

Although the Navy Facilities Command is already fielding a replacement for the current system, named electronic Family Housing (EFH), FAMIS was operational at the time of this study. The Navy's housing management believes that "the interests of the Navy family housing community are best served by deploying a single, standard system to all housing activities throughout the Navy" (Harris, 2001, p. 12).

F. SUMMARY

The housing management system used at NSAMB is incomplete, unreliable, burdensome, and frustrating to use. The HWC counselors spend too much time switching between applications and files, resulting in transposition errors in data entry and user frustration. Online housing applications are lost by the system. In addition, there are so many patches to make the current applications work, that there is no single, intuitive menu to meet their requirements. They need an application that can import their

intuitive menu to meet their requirements. They need an application that can import their existing data, share tables with an Internet server, display user-friendly and intuitive forms and reports, and identify critical information to assist decision makers. A new application can import the existing data files and spreadsheets into a relational database management system, where users can access the information more effectively and efficiently.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PROPOSED HOUSING SYSTEM

A. INTRODUCTION

In order to elicit requirements for the new system, we have chosen to use the Knowledge Analysis of Tasks (KAT) method as an aide. KAT is an object-oriented task analysis method proposed by Peter Johnson. While KAT is not a requirements elicitation method per se, it provides a process to help describe the domain, understand the objects and interactions, and construct a model of the housing assignment and termination process. KAT is a method:

> ...concerned with collecting data from variety of sources (e.g., protocol analysis, standard procedures, textbooks, interviews), analyzing these data to identify individual elements involved in the task (e.g., objects, actions, procedures, goals, and subgoals), and constructing a model of the overall knowledge used by people accomplishing the task of interest (Bruegge, 2000, p. 120).

KAT differs from the Unified Modeling Language (UML) in two ways: what is does and when it is done. KAT is a knowledge analysis method that provides a process for requirements elicitation, while UML is a modeling language, and not a method (Fowler, 1997, p. 43). Second, KAT occurs before we apply UML. Software designers apply KAT in the requirements elicitation phase prior to using UML in the requirements elaboration phase. We will apply the KAT methodology in order to understand the requirements before using UML diagrams to express the requirements.

In order to analyze the tasks involved in the assignment and termination process, we collected information from various sources. First, we interviewed several Housing Office counselors, including the division supervisor, and users of FAMIS in other divisions of housing, such as maintenance and facilities. We asked the users to describe

17

problems with the current system, as well as useful features and capabilities of the system. In some cases, we went through actual scenarios in which the current system fails to support housing needs, and discussed possible solutions.
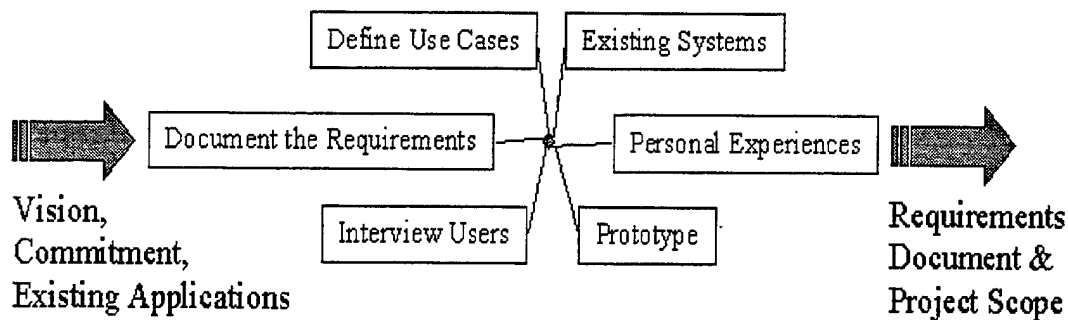
Figure 3-1. Sources of Information for Requirements Gathering (After Ambler, 2000, p. 67)

Another source of information is personal experiences with the assignment and termination process. I have lived in government quarters twice and have received referral services more than a few times. During the course of this thesis, I arranged an assignment for a friend and paid particular attention to the information requirements and interactions with the occupant, the counselor, and the system. I recently filed my own "intent to vacate" and personally experienced the shortcomings of the current system.

Other sources of information to determine requirements include the current system and documentation, as well as the NSAMB housing regulations. The current system acts as a prototype because it allows users to interact with the system and provides feedback for making improvements. The current system documentation contains a data dictionary and helps explain the functionality. Finally, the NSAMB housing regulations provide the key business rules and constraints that shape the use cases, and ultimately, the requirements.

18

B. FUNCTIONAL REQUIREMENTS FOR THE HOUSING ASSIGNMENTS AND TERMINATIONS SYSTEM (HATS).

In order to determine the functional requirements for the new system, named the Housing Assignment and Terminations System (HATS), we analyzed the assignment and termination process from the various sources, deduced the required functions, created a functional decomposition, and captured the use cases to refine the requirements.

HATS is a computer information system that assigns eligible DoD employees to military family housing and provides off-base referrals. The system allows users to enter data, modify existing records of occupants and housing units, view and print reports and DOD forms, and use decision support tools to help management make better decisions.

HATS consists of a database server and client terminals running a database application. HATS is also accessible to users through an Internet browser. The main Internet feature is to allow incoming service members to apply for housing (and enter their information directly into the database, without the assistance of an HMS). Another Internet feature is to allow incoming service members to view and print the waiting list. Internet users can only view the reports—they cannot modify or delete files.

With slight modification, HATS can be used at any U.S. DOD housing office in the world.

C. NONFUNCTIONAL REQUIREMENTS

Some requirements are important to the look, feel, and performance of the system, but they don't provide any additional functionality to the system.

1. Ease of Use

The system must be easy to navigate between functions and it must use drop down menus whenever possible. The GUI should be appealing and navigation through the menus should be easily accomplished with either a keyboard or a mouse. Users should not reach dead end screens; instead, they should be a mouse click or two away from important functions. The design should be similar to popular, useful Web sites that offer easy navigation.

2. Scalability

The NSAMB HWC only services about 5,000 customers currently; however, it's possible that DOD could increase the number of service members in the area. In addition, it would be an added benefit to DOD to be able to use the NSAMB HATS for another military installation that has over 10,000 customers. In both cases, the system should be able to handle the increase of records without suffering performance shortfalls.

3. Availability

The system has to be available 24 hours a day, 7 days a week because the customers live in nearly every time zone.

4. Responsiveness

We personally observed a task at the HWC that took over 45 seconds using the current system. The proposed system should respond to requests in less than 15 seconds.

D. PSEUDO REQUIREMENTS

1. Documentation

The Housing Office requires a user's manual to explain the system's details, such as a menu summary, valid field options, and a glossary.

2. Hardware and Software Considerations

The system must be compatible with IT-21 hardware and software. Specifically, the TRAC-Monterey requires that the system be designed using a Microsoft Access database, since the software is already required by Navy Housing Offices around the world. In addition, the system must interface with the Housing Office's Web server, which runs Microsoft's Internet Information Server (IIS) software.

3. Error Handling and Extreme Conditions

When encountered, the system should gracefully notify the user of errors or other adverse conditions and offer a solution. Whenever possible, the forms and menus should only offer valid input ranges to avoid errors.

4. Physical Environment

The system has to operate within the constraints of the current physical environment, meaning that it has to work with the existing hardware and software at the Housing Office.

5. Security Issues

The system must enforce access control using user identification and passwords. Once the system allows access to users, there is no need for further compartmentalized security. The system interface with the Web server is particularly vulnerable to unauthorized access and malicious activity.

6. Resource Issues

Because there will be as many as 20 concurrent users on the system, it must resolve competing reads and writes to the database. The system has to either lock records, or specific tables, to ensure data consistency and to avoid deadlock.

E. SYSTEM MODELS

1. Functional Decomposition

The functional decomposition is a process that starts with defining the top-level system functions, each of which is further refined in iterative steps. The end result is bottom-level functions that are easily captured in pseudo code and eventually programmed in a computer language or using an application development suite, such as Microsoft Access.

2. Use Case Model

Use cases are primary elements in object oriented software development. Using the requirements derived from the functional decomposition and the KAT, combined with the functions of the current system and user feedback, we developed 21 use cases that describe the interaction between the users and the system. For each use case, we used a common template, shown in Table 3-1, that includes 12 sections.

Using the template shown in Table 3-1, we used an iterative process to complete the template for each of the 21 use cases, which are summarized in Table 3-2 and listed in entirety in Appendix A.
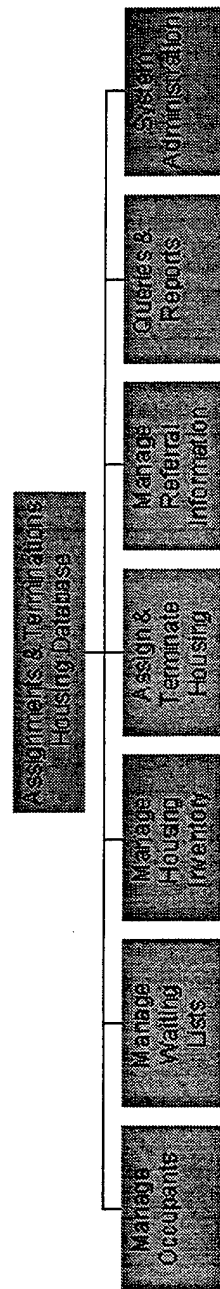
Figure 3-2.   Top-level Functions of the Proposed System
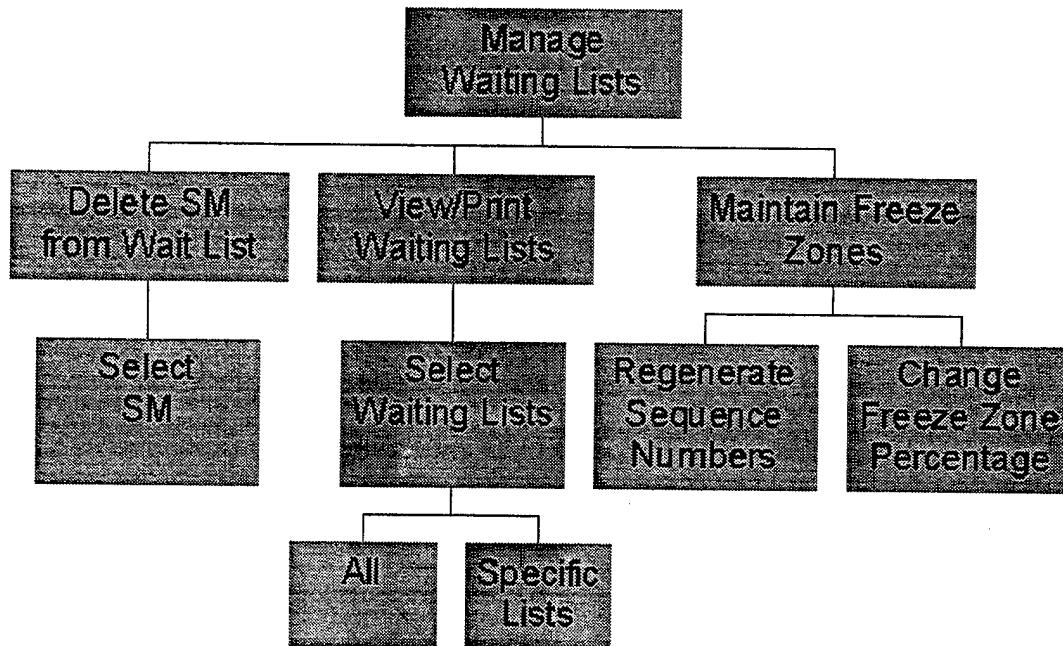
Figure 3-3.   Decomposed Second Level Functions

Use case name	
Participating actor	
Summary	
Basic Course Of Events	
Alternative Paths	
Exception Paths	
Triggers	
Assumptions	
Preconditions	
Postconditions	
Related Business Rules	
Special Requirements	

Table 3-1.   Use Case Template

UC 5	Apply Online	The SM visits the NSAMB Housing Web site. S/he fills out the DD Form 1746, Application For Assignment To Housing, by entering personal information in the required fields of the Web page.
UC 10	Apply For Housing	The SM applies for housing in person, or through the mail. She fills out the DD Form 1746, Application For Assignment To Housing, and gives a hard copy to the HMS.
UC 15	Modify SM	The HMS may need to correct inaccurate or outdated SM information such as number of dependents or an incorrect SSN.
UC 20	View and Print Waiting Lists	The Housing Office constantly views the waiting list to update SMs on their place on the list and their expected availability date. The HMS's also print periodic lists to display in the Housing lobby.
UC 25	Move SM to Actual Waiting List	The SM gives the HMS verification of dependents, orders, and departure from last assignment. With this information, the HMS is authorized to move the SM to the actual waiting list.
UC 30	Promote SM	The SM receives a promotion since she entered the expected waiting list, making her eligible for a higher category of housing. The HMS moves the SM to the new actual waiting list.
UC 35	Delete SM From Waiting List	A SM is on either the expected or actual waiting list, but has decided to live off base.
UC 40	Maintain Freeze Zone	Housing stabilizes the position of the top ten percent of SMs on each actual waiting list by not placing new arrivals in the freeze zone, regardless of rank, duty assignment, or detachment date. They make exceptions for key and essential SMs, based on specific guidance. When this happens, Housing places these SMs at the top of the freeze zone, without displacing SMs with a housing offer.
UC 45	Offer Housing	The HMS notifies a SM that her sequence number is next on the actual waiting list. The system finds three houses matching the SM's

		category and the HMS offers the SM to choose one house to live in (after visual inspection).
UC 50	Make Assignment	The SM chooses to occupy one of the houses offered by the HMS, who makes the official assignment (matching the SM with a housing address).
UC 55	Terminate Assignment	SMs frequently receive orders to move to other duty stations. Once a SM has orders, she completes an "Intent to Vacate" letter and informs the HMS of the specific dates. Eventually, the SM moves and terminates the housing assignment.
UC 60	Prepare CRL	An incoming SM wants information on off-base housing. The SM gives the HMS search criteria for desired living arrangements. The HMS enters the information into HATS and prints a Customer Referral List for the SM.
UC 65	Add or Edit MFH	Occasionally, the HMS needs to edit the description of MFH to reflect upgrades or modifications. In the event of new construction, this is the use case used to add the address to the inventory.
UC 70	Add or Edit Private Property	Details about off base rental property change almost daily. The HMS needs to add new rental property to the system and change descriptions and terms to existing properties.
UC 75	Delete Address	For one reason or another, the HMS needs to remove either government or privately-owned property. For example, the owner of an off base property decides to sell and no longer offer it for rent. Or in the case of the government quarters, the base sells the property to another entity.
UC 80	Add or Edit Owner or Manager	The HMS adds or edits information about an owner or property manager. This can happen due to a sale, or when a new manager takes over the property.
UC 85	Change Availability	When a specific house is unavailable to SMs, the HFS has to provide a date when she thinks the unit will be ready for occupancy. Once move-out or other maintenance is complete, the HFS notifies the system that the MFH is

		fit for a new occupant.
UC 90	View and Print Reports	There are over 20 reports that users request from the system on any given day. The HMS or HFS can use the "Queries & Reports" function of HATS or s/he can use menus in the other functions to view and print reports.
UC 95	Manage Accounts	Employees come and go, get promoted, or change positions at the Housing Office, requiring changes to user access to the system.
UC 100	Maintain Database	The HBDA needs to maintain the database in the system occasionally. S/he may need to change forms, reports, or other internal files in order to maintain the system. Also, Housing does not want another user inadvertently changing the database structure and corrupting the system.
UC 105	Authenticate User	All users who request entry to the system must prove they are who they say they are (authentication). This applies to users inside Housing as well as outside. Who the users are determines what they have access to (authorization).

Table 3-2.   Summary of Use Cases

A use case diagram is an essential requirement in the UML. It helps system designers visualize the use cases and understand the scope of the system. Figure 3-4 depicts the UML use case diagram for the proposed system.
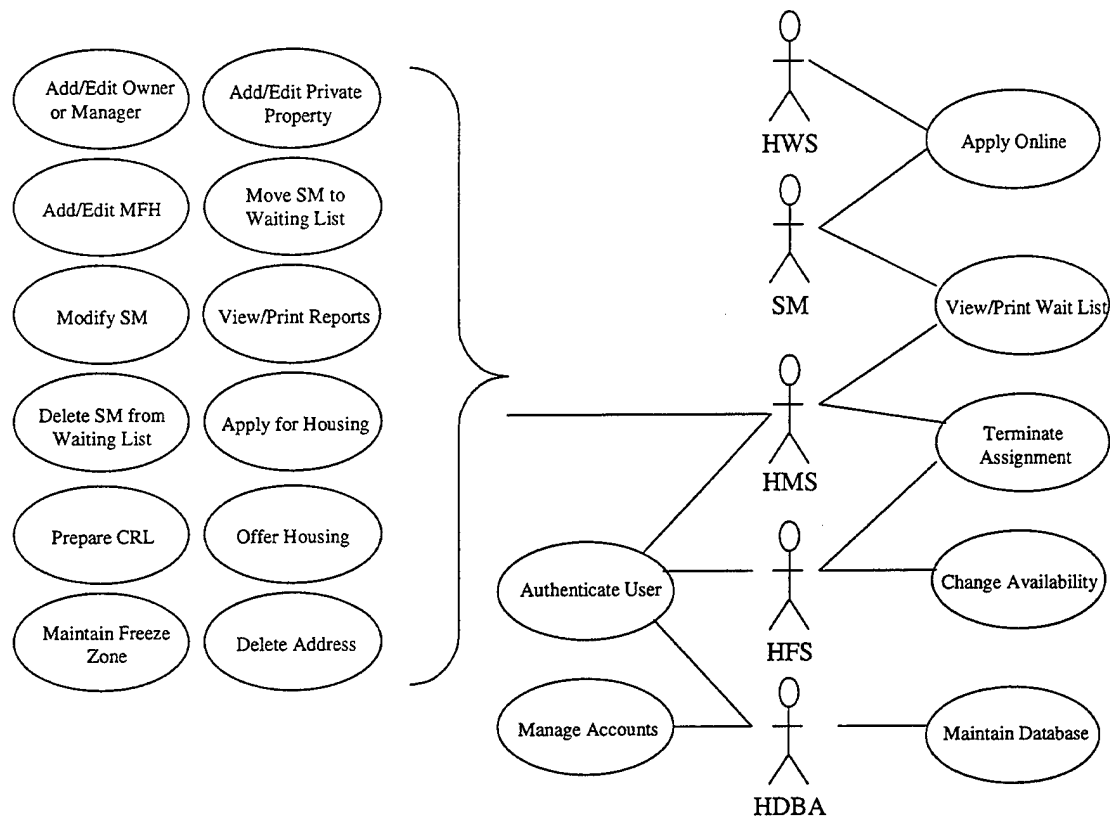
Figure 3-4. HATS Use Case Diagram

F.   SUMMARY

Defining requirements is a critical step in designing information systems. Too many times, system engineers make mistakes in the requirements analysis phase that cost the project time, money, and sometimes, critical functionality. The Knowledge Analysis of Tasks (KAT) method helps describe the problem domain and objects and interactions.

Using the KAT methodology, we analyzed various sources of information before determining the system requirements and expressing them using a UML diagram. The functional decomposition and use cases were the most useful tools for determining requirements for the Housing Assignments and Terminations System (HATS).

IV. HOUSING SYSTEM PROTOTYPE DESIGN

A. OVERVIEW

After determining the requirements for the Housing Assignments and Terminations System, we can begin the design phase of our HATS prototype. The importance of this phase cannot be overstated. Mistakes in design will most likely cascade into all subsequent phases, with potential serious effects upon database performance. We start with a logical design, using a standard database modeling application. Next, we transform the model into a physical design of the database. Once our design is complete, we will must extract the data from the HWC's legacy system (FAMIS) and Access database, and fit the data with our schema.

B. LOGIGAL DATABASE DESIGN

In our requirements analysis, we identified use cases that depict interaction between users and the system. As described in the use cases, the users typically create, replace, update, or delete information in order to complete forms and create reports about service members, houses, referral property, and other entities in their environment. We decided to model these entities, also known as objects or semantic objects, using the semantic object model (SOM) shown in Figure 4-1.

A semantic object is a "named collection of attributes that sufficiently describes a distinct identity" (Kroenke, 2000, p. 74). A service member is an example of a semantic object because it describes a distinct identity that we wish to track and report. An attribute of an object is a characteristic that describes the object in more detail. For example, the attribute "LastName" describes the object SERVICEMEMBER.
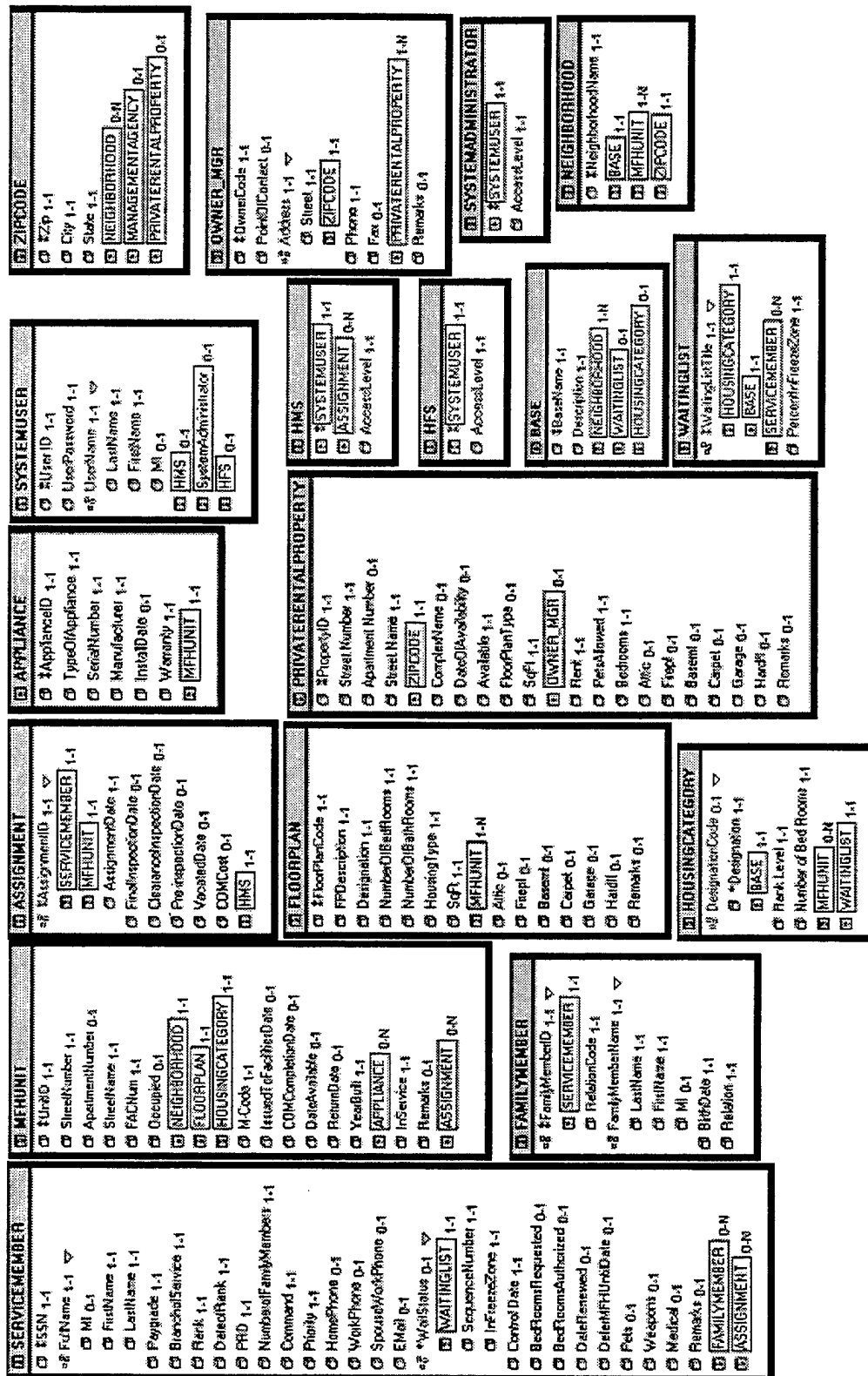
Figure 4-1.   Semantic Object Model for HATS

32

The first object shown in Figure 4-1 is the housing customer, the SERVICEMEMBER. The unique identifier for each service member is his or her social security number (SSN). Each service member in our model has one-and-only-one of the following attributes: FullName, Paygrade, BranchOfService, Rank, DateOfRank, PRD, NumberOfFamilyMembers, Command, Priority, and ControlDate. A service member may have zero-to-one of the following: HomePhone, WorkPhone, SpouseWorkPhone, Email, WaitStatus, BedroomsRequested, BedroomsAuthorized, Pets, Weapons, Medical, Date Renewed, and DeferMFHUntilDate. If a service member has a WaitStatus, s/he must have one-and-only-one of the following: WAITINGLIST, SequenceNumber, and InFreezeZone. A service member may have zero-to-many FAMILYMEMBERs and ASSIGNMENTs. Because the SERVICEMEMBER object is related to other objects, such as FAMILYMEMBER, the model has to capture the appropriate relationships in order to have any meaning. In Figure 4-1, related objects are enclosed in rectangles with their associated cardinalities appearing to the right. APPENDIX B contains a description of each of the remaining objects and their relationships.

C.   PHYSICAL DATABASE DESIGN

Although data modeling is an important task in database development, a model by itself is only a means to an end. We have to transform the semantic object model into relational tables in a database which will be used to use the data in a meaningful way. We will use a three-step process to transform the SOM into a relational database design:

- Transform the SOM into Relations
- Normalize Relations
- Create the Database Schema

Once the model is transformed into a relational database design, we can migrate the data from the current system into the proposed system and focus on fulfilling the requirements for the system.

1.   Transforming the SOM into Relations

We follow algorithms proposed by Kroenke (2000, pp. 169-185) for transforming the semantic objects into relations, or tables. The SERVICEMEMBER object, shown in Figure 4-2, serves as an example of implementing transformation algorithms. The first step is to determine the type of object. In this case, SERVICEMEMBER is a hybrid object because it contains at least one multi-valued group attribute (WaitStatus) that includes an object attribute (WAITINGLIST). The recommended approach for this type of object is to apply a combination of techniques for composite and compound objects. A composite object is an object that contains one or more multi-valued, non-object attributes. A compound object contains at least one object attribute.

The next step is to create a relation for SERVICEMEMBER and both of the contained objects FAMILYMEMBER and ASSIGNMENT. The primary key for the SERVICEMEMBER table is the social security number (SSN) attribute. Since a service member may have many family members, the primary key for the FAMILYMEMBER table consists of the two attributes: the foreign key attribute SSN (from SERVICEMEMBER), and the RelationCode attribute, which specifies a unique relation within a particular family. The ASSIGNMENT table primary key consists of two foreign keys and one local key: SSN (SERVICEMEMBER), UnitID (MFHUNIT), and AssignmentDate. Since a service member can have zero-to-many family members, we place the key of the SERVICEMEMBER table (SSN) as a foreign key attribute in the

34

FAMILYMEMBER table. In this case, the foreign key becomes part of the primary key of the FAMILYMEMBER table because a FAMILYMEMBER instance will be related to one and only one SERVICEMEMBER instance. Similarly, the ASSIGNMENT object associates a service member with a house (MFHUNIT), so it too is a special case of a compound object known as an association object. In this case, we again place the key (SSN) for SERVICEMEMBER as a foreign key in the ASSIGNMENT relation. As with the FAMILYMEMBER, this foreign key also serves as part of the primary key for the ASSIGNMENT table. No attributes of FAMILYMEMBER or ASSIGNMENT are placed in the SERVICEMEMBER relation.

Next, we apply the technique for composite objects to transform the multi-value group, WaitStatus. Since this group contains another object, WAITINGLIST, we establish a relation for WAITINGLIST. Because a WAITINGLIST can have many service members awaiting housing, we place the keys of WAITINGLIST (Designation and BaseName) in the SERVICEMEMBER relation as foreign keys. Finally, we create a relation for the group identifier, WaitStatus. Since the relationship between the new relation WAITSTATUS and SERVICEMEMBER is one-to-one, we place the key of one relation into the other as a foreign key. Now, all composite and compound objects are transformed into the relations, leaving only simple attributes.

The final step in transforming the SERVICEMEMBER object is representing the simple attributes, such as Paygrade, BranchOfService, and others shown in Figure 4-2. We then place the simple attributes in the SERVICEMEMBER relation and the table design, shown in Table 4-1, is complete.

35

SERVICEMEMBER
- ‡SSN 1-1
- FullName 1-1 ▽
  - MI 0-1
  - FirstName 1-1
  - LastName 1-1
- Paygrade 1-1
- BranchofService 1-1
- Rank 1-1
- DateofRank 1-1
- PRD 1-1
- NumberofFamilyMembers 1-1
- Command 1-1
- Priority 1-1
- HomePhone 0-1
- WorkPhone 0-1
- SpouseWorkPhone 0-1
- EMail 0-1
- *WaitStatus 0-1 ▽
  - WAITINGLIST 1-1
  - SequenceNumber 1-1
  - InFreezeZone 1-1
- Control Date 1-1
- BedRoomsRequested 0-1
- BedRoomsAuthorized 0-1
- DateRenewed 0-1
- DeferMFHUntilDate 0-1
- Pets 0-1
- Weapons 0-1
- Medical 0-1
- Remarks 0-1
- FAMILYMEMBER 0-N
- ASSIGNMENT 0-N

ASSIGNMENT
- ‡AssignmentID 1-1 ▽
  - SERVICEMEMBER 1-1
  - MFHUNIT 1-1
  - AssignmentDate 1-1
- FinalInspectionDate 0-1
- ClearanceInspectionDate 0-1
- Pre-inspectionDate 0-1
- VacatedDate 0-1
- COMCost 0-1
- HMS 1-1

WAITINGLIST
- ‡WaitingListTitle 1-1 ▽
  - HOUSINGCATEGORY 1-1
  - BASE 1-1
- SERVICEMEMBER 0-N
- PercentInFreezeZone 1-1

FAMILYMEMBER
- ‡FamilyMemberID 1-1 ▽
  - SERVICEMEMBER 1-1
  - RelationCode 1-1
- FamilyMemberName 1-1 ▽
  - LastName 1-1
  - FirstName 1-1
  - MI 0-1
- BirthDate 1-1
- Relation 1-1

Figure 4-2.   SERVICEMEMBER, ASSIGNMENT, WAITINGLIST, and
FAMILYMEMBER Objects

36

Relation	Attributes
SERVICEMEMBER	(SSN, LastName, FirstName, MI, Paygrade, BranchOfService, Rank, DateOfRank, PRD, NumberOfFamilyMembers, Command, Priority, HomePhone, WorkPhone, SpouseWorkPhone, Email, ControlDate, BedroomsRequested, BedroomsAuthorized, DeferMFHUntilDate, Pets, Weapons, Medical, Remarks, Designation_FK, BaseName_FK)

Table 4-1.   Transformed SERVICEMEMBER Relation

We continued the transformation process by applying Kroenke's algorithms to the remaining semantic objects in Figure 4-1. Appendix C shows all of the transformed relations.

2.   Normalization

Once we have established our relations, we need to normalize any undesirable relations to avoid modification anomalies, which are unwanted consequences in our relations. The purpose of normalizing our relations is to obtain a refined relational model that performs according to our design.

A normal form defines a class of relations "and the techniques for preventing anomalies" (Kroenke, 2000, p. 119). Depending on a relation's structure, it can be in one of six normal forms: first, second, third, Boyce-Codd, fourth, and fifth.

All of the tables of data in Appendix C are in first normal form because they meet the following conditions:

- All cells of the table are of a single value;
- All attributes are of the same kind;
- Each column has a unique name;
- No two rows are identical.

In order to be in second normal form, all non-key attributes must depend on the entire key. For relations with a single attribute as key, they are automatically in second normal form. Several of our relations meet this requirement because they have a single attribute as the key: MFHUNIT, FLOORPLANCODE, APPLIANCE, BASE, ZIPCODE, NEIGHBORHOOD, PRIVATERENTALPROPERTY, OWNER, SYSTEMUSER, HMS, HFS, and SYSTEMADMINISTRATOR. All of the relations with multiple attribute keys are in second form because the non-key attributes are about the entire key, not just part of the key.

A relation is in the third normal form if it is in already in second normal form and its attributes are about just the key, not about the key and another non-key attribute. Even when a relation is in the second normal form, we can still experience modification anomalies. One kind of modification anomaly that we want to avoid is a deletion anomaly. A deletion anomaly occurs when we delete facts about one entity and inadvertently delete facts about a separate entity (Kroenke, 2000, p. 118). For example, in the SERVICEMEMBER relation, a service member, like John Doe, has an SSN (123-45-6789), a Paygrade (O6), and a Rank (colonel). In addition, assume that John Doe is the only instance of SERVICEMEMBER with the rank of colonel in the relations. If we delete John Doe's record from our database, we also inadvertently delete the fact that a

service member in the Paygrade of O6 has the rank of colonel. The solution to this anomaly is to create two separate relations, such as the ones shown in Table 4-2. Now, the SERVICEMEMBER relation is in the third normal form.

Relation	Attributes
SERVICEMEMBER	(SSN, LastName, FirstName, MI, Paygrade, BranchOfService, DateOfRank, PRD, NumberOfFamilyMembers, Command, Priority, HomePhone, WorkPhone, SpouseWorkPhone, Email, BedroomsRequested, BedroomsAuthorized, DeferUntilDate, ControlDate, Pets, Weapons, Medical, Remarks, Designation_FK, BaseName_FK)
RANK	(Paygrade, Rank)

Table 4-2.   Creating Two Separate Tables to Avoid Deletion Anomalies

Since this prototype is not intended for full implementation at the HWC, we feel that normalizing the relations to the third normal form is sufficient for this purpose. All tables in Appendix C now satisfy 3rd normal form, and are ready to be put into the database schema.

3.   Creating the Database Schema

The schema defines the database's structure, tables, relationships, domains and business rules (Kroenke, 2000, p. 30). We formalized part of the schema when we created the semantic object model using the Cyberprise Tools DBApp Developer 2.5 Standard Edition (by Wall Data, Inc.). As we defined the objects, we also defined the attribute domains. We then used the "Create Database" tool in DBApp to create our initial MS Access database, complete with the tables and data definitions that are part of our schema.   After translating the schema into Access, the database is now ready to receive the HWC's operational data.

D. SUMMARY

The design phase of our HATS prototype started with an understanding of the requirements. We created a semantic object model to represent our proposed system. Using DBApp, we captured the foundation of the schema in our SOM. Then, using a three-step process, we transformed the object model into a physical design, represented by our relational model. Now that our design phase is complete, we need to extract the pertinent data from the HWC's systems, clean it, and merge it into our Access tables.

V.   DATA MIGRATION

A.   INTRODUCTION

Migrating data from one system to another sounds like a simple task; however, it can be a difficult process for several reasons. First, the people performing the migration have to thoroughly understand the existing database schema; otherwise, they may commit errors that have detrimental consequences when applied to the new system. In many cases, the original design team is not part of the migration, which may occur well after the initial system was built. In our case, 15 years elapsed since the original system was implemented and no one from the original design team was available. Second, the existing schema may not match the newly designed schema. In our case, the HWC uses many flat files, with data definitions that are not consistent with current "best practices". In order to convert the flat files, we may have to perform detailed data manipulation, and in the worst case, re-enter some of the data to conform to our design. Finally, the people performing the data migration have to understand the business process, rules, and constraints that were in place during the design and implementation of the legacy database as well as those currently in use by the organization.

Planning a migration for the HWC's data is further complicated because of the recent history of its military customer in the Monterey Bay area. Over the past 10 years, the HWC has recorded information about the housing areas in several tables, using different aliases, and creating inconsistencies in the database. For example, in one table, the neighborhood describing a particular unit appears as POM, while the same field is called PA in another table. This confusion is partially a result of a U.S. Defense Department program, known as Base Realignment and Closure (BRAC), designed to cut

military infrastructure costs by closing American bases around the world. Today, the HWC serves the entire military community in Monterey. However, just ten years ago, the Army's former Housing Office at Fort Ord managed over half of the current housing inventory. The NSAMB assumed the responsibility to manage military housing at the Presidio of Monterey (also formerly managed by the Army) and Fort Ord. Both former Army housing areas are defined using multiple naming conventions in both the FAMIS and Access tables. Over the years, Fort Ord's name has changed several times. For a few years, it was known as the Presidio of Monterey Annex, abbreviated as PA, or POMANNEX. Recently, it was renamed Ord Military Community (OMC). In addition, the Presidio of Monterey, abbreviated POM, also goes by the name DLI, which is short for the major command stationed at the Presidio, the Defense Language Institute. Our relational prototype must raise the level of data integrity and allow instant, cascading updates of such information with few keystrokes.

B. DATA EXTRACTION AND MIGRATION

1. Overview

The HWC provided access to all their database files, including the FAMIS legacy database files and their MS Access database files. We reviewed the tables and talked to the HWC database administrator to make sure that we understood the schema and its link to their business process. Then, we analyzed the existing tables and performed a data crosswalk, as shown below in the SERVICEMEMBER table migration (Table 5-1), for each of our relations, matching the existing schema with our prototype schema whenever possible. In some cases, the table attributes matched and we only needed to import the files using the MS Access menus. In the majority of cases, however, the tables were

either partially matching, or not matching at all. Making matters worse, the HWC's Access database did not have primary keys for the tables and the relations were not linked or related at all. Thus, we had to deal with flat files, many of which we could not directly link to the other tables in FAMIS, which contained most of the operational data. Many tables contained errors in at least one of these categories:

- Incomplete or null column values;
- Inconsistent naming conventions;
- Rows not identified by unique key(s);
- Redundant columns (as well as tables);
- Contradictory values between tables;
- Obsolete columns—not supported by current rules, policies, and constraints.

In order to maintain a high degree of data quality, we had to perform lengthy data preparation and migration steps. In the following discussion, we explain some of the challenges we faced preparing the legacy database files, and eventually migrating them to our prototype.

2. SERVICEMEMBER Table

For all of the tables, we prepared a data crosswalk to determine which tables and attributes in the existing database files would migrate into our tables. Table 5-1 shows the data crosswalk for the SERVICEMEMBER table.

COLUMN TO FILL	SOURCE TABLE	SOURCE COLUMN
SSN	ASPRS (FAMIS)	AS_SSNO
LastName	ASPRS	AS_LNME
FirstName	ASPRS	AS_FNME
MI	ASPRS	AS_MNME
Paygrade	ASPRS	AS_PGRD
BranchOfService	ASPRS	AS_BSER
DateOfRank	ASPRS	AS_RNKD
PRD	ASPRS	AS_PRDT
NumberOfFamilyMembers	ASPRS	AS_DEPD
Command	ASPRS	AS_COMD
Priority	ASPRS	AS_PRIO
HomePhone	ASPRS	AS_HMEP
WorkPhone	ASPRS	AS_WRKP
SpouseWorkPhone	ASPRS	AS_SWRP
Email	* (No Match)	
BedRoomsRequested	ASPRS	AS_BEDS
BedRoomsAuthorized	ASPRS	AS_BEDR
DeferMFHUntilDate	ASPRS	AS_DDRF
ControlDate	ASWTLD (FAMIS)	AS_CDTE
Pets	* (No Match)	
Weapons	* (No Match)	
Medical	ASPRS	AS_HAND
Remarks	ASPRS	AS_RMKS
Designation_FK	HIUNT (FAMIS); LAMHIST, PROJECTION HISTORY (HWC)	HI_UNTP Desig#, Res-cat
BaseName_FK	HIUNT (FAMIS); LAMHIST, PROJECTION HISTORY (HWC)	HI_PNME Park

Table 5-1.   SERVICEMEMBER Table Data Crosswalk

The crosswalk shows that many of the attributes in our SERVICEMEMBER table match attributes in the FAMIS table, ASPRS. However, we also see that the existing database has no attributes describing a service member's Email address. Neither does it track whether or not s/he will have pets or weapons in MFH. Therefore, these attributes will contain null values initially; however, once we implement the prototype and create new records, we will be able to track these attributes. The last two attributes, Designation_FK and BaseName_FK, are both foreign keys that must be extracted from multiple tables, after we migrate more data into our prototype.

After migrating data from the ASPRS table, we checked for duplicate records. Access 2000 provides a convenient wizard, shown in Design View in Figure 5-1, to query tables for duplicate records. Using this feature, Access reported finding over 30 duplicate records, summarized in Table 5-2, mostly resulting from partially completed records, or multiple housing assignments. The FAMIS database did not recognize the SSN attribute as unique, allowing users to enter duplicate records for service members. We deleted the duplicate records with the poorest data quality. For example, in Figure 5-1, we deleted the first record for Christopher Lehner because it did not list his MFH, while the second record did. Now, we ran the query again. Access returned a null report, allowing us to set the SSN as the key for the SERVICEMEMBER table.
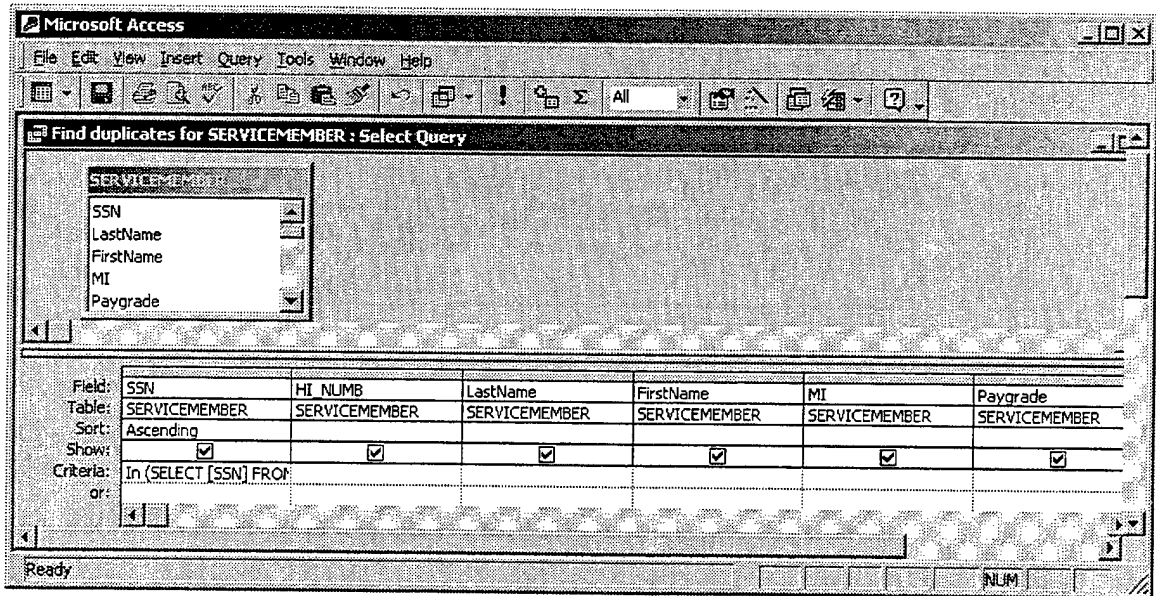
Figure 5-1.   The Design View for the "Find duplicates" Query

SSN	StreetNumber	LastName	FirstName	MI	Paygrade
153681111		LEHNER	CHRISTOPHER		O3
153681111	3023	LEHNER	CHRISTOPHER		O3
425272222		LEWIS	ANGELA	M	E5
425272222	2943	LEWIS	ANGELA	M	E5
602193333					
602193333	600	KAMARAS	KONSTANTINOS		O2

Table 5-2.   Query Results for Duplicate Records

3.   MFHUNIT Table

The MFHUNIT table represents the military houses and apartments managed by the HWC. Again, we performed a data crosswalk and mapped out the attributes. Since we planned on using most of the attributes from the FAMIS table, HIUNT, we decided to import the table, change the data definitions to match our design, and save the table under the new name, MFHUNIT. After importing the HIUNT table from the FAMIS database,

46

we used the MS Access design view, as shown in Figure 5-2 below, to implement the data crosswalk and create the MFHUNIT table. The column Field Name displays the initial MFHUNIT attribute names, Data Type displays the domain, and Description contains our input for the new attribute names. We deleted the few fields that we did not need from our table, such as HI_0107. Upon closer examination of the data, however, we noticed serious data inconsistencies and other potential problems.

Many of the FAMIS and Access tables contained null values in columns that should have been complete. We noticed that about 10% of the records in the FAMIS table HIUNT contained null values for the building number (facility number) column, even though other tables, such as the Access POMAFAC table, contained the correct information. For example, the unit located at 302 Ardennes Circle did not have a building number in the table, however over 90% of the records had an entry. We found the correct building number in the POMAFAC table under the FAC# column, and later created a query to update the values.

Another problem with the data was misspelled words. For example, a street name in the HWC's history table for OMC, ANNHIST2, was misspelled "BOUGAINVILL". In order to get the addresses to a common definition, we performed queries to add the street name suffixes. When we searched for the correct spelling "BOUGANVILLE", we did not find any units on that street. After finding the mistake, we were able to change the name to "BOUGANVILLE ROAD". This kind of error will not occur in our prototype because we will use drop down lists and validation rules in Access.

Figure 5-2.   MS Access Design View of the Housing Inventory Table, HIUNT

Unfortunately, the Housing Office Access tables had many inconsistencies that made the preparation and migration process extremely tedious and time-consuming. Because FAMIS was neither flexible nor user friendly, the HWC created two other tables in an MS Access database, POMAFAC and LA-FAC, to track other information about MFHs. The Access and FAMIS tables, summarized in Tables 5-3 through 5-5, did not have compatible attributes with which to relate the tables. We had to manipulate the one common attribute type, address column, in order to get them on a common naming convention, thus making unique keys that we could then join in an Access query. For example, the FAMIS table HIUNT used the key HI_NUMB to distinguish unique units. The key for the POMAFAC table, used to track OMC neighborhoods, was the FAC-REC# column, which was not compatible with the HI_NUMB column, nor any other columns in the HIUNT table. The key for LA-FAC was also named FAC-REC#, which we eventually matched with the unique identifier in HIUNT, the HI_NUMB column.

By this point, we had recorded over five different address-naming conventions that prohibited us from relating several tables, including MFHUNIT. In addition to the address-naming conventions shown in Tables 5-3 through 5-5, other database tables contained addresses with spaces and dashes separating the street names, numbers, and apartments, such as "372-C Bergin Drive" and "372 C Bergin Drive". Another problem, exhibited in Table 5-5, is that in some tables, the address components were placed in one field, while the other tables split the apartment, street number, and street name into separate fields. We created several queries in Access that manipulated the fields, either concatenating or separating the address components. Then, we created a lookup table,

49

shown in Figure 5-3, to query and update the addresses to calibrate them to the same standard.

HI_NUMB	HI_SNAM	HI_SNUM	HI_APTN
1	BERGIN DRIVE	370	A
10	BERGIN DRIVE	371	B
100	BERGIN DRIVE	387	C
101	BRITTANY ROAD	415	
102	CORREGIDOR ROAD	166	
103	BIDDLE LANE	3	
104	BIDDLE LANE	4	

Table 5-3.   Summarized FAMIS Unit Inventory Table, HIUNT

No#	STREET	FAC-REC#
300	AACHEN ROAD	P06544
301	AACHEN ROAD	P06530
302	AACHEN ROAD	P06543
303	AACHEN ROAD	P06531
304	AACHEN ROAD	P06542
305	AACHEN ROAD	P06532
306	AACHEN ROAD	P06541A

Table 5-4.   Summarized Housing Office Unit Inventory Table, POMAFAC (OMC
Neighborhood)

FAC-REC# .	RES ADD
1	370 A BERGIN DRIVE
2	370 B BERGIN DRIVE
3	370 C BERGIN DRIVE
4	370 D BERGIN DRIVE
5	370 E BERGIN DRIVE
6	370 F BERGIN DRIVE
7	370 G BERGIN DRIVE

Table 5-5.   Summarized Housing Office Unit Inventory Table, LA-FAC (La Mesa
Neighborhood)

50

Figure 5-3.　　Address Lookup Table

Another address naming convention problem involved tables containing street suffixes, such as Drive or Street, which other tables did not. Again, the address was the only common attribute name that we could use to relate some of the tables, so we had to find a way to standardize the addresses across the tables. We created a lookup table, shown in Figure 5-4, that we used in update queries to modify the street names.

51

Figure 5-4.   Street Name and Suffix Lookup Table

4.   ASSIGNMENT Table

The assignment history table in FAMIS, ASHDT, was missing one of the most important attributes in many records—assignment dates. We designed the table such that the key included the date of assignment to MFH, however, some of the records had a null value in the field. Access will not allow key columns to contain null values. We had to search the database for such records and determine the most likely assignment date, based on other clues, such as the application date and the vacate dates.

Unfortunately, in some records, the vacate date was also null. Therefore, we had to search other database tables, like the SERVICEMEMBER table, to find any information that would help fill our table.

52

HI_NUMB	AS_SSNO	AS_ASGN	HI_SNUM	HI_SNAM	AS_LNME
3665	306521212		196	NORMANDY ROAD	SMITH
2807	323842323		224	BUNA ROAD	VANDEVENTER
3378	238604545		203	NIJMEGEN ROAD	MAENG
4005	597865656		155	OKINAWA ROAD	MCNEIL

Table 5-6.   Assignment History Table with Null Values in the AS-ASGN Column
(Date of Assignment)

We realized that this detective work was error-prone, especially in the case where

the HI_VACT field was also null (and therefore we could not estimate the assignment

date based on the vacate date). However, our main goal was to create a key, even if the

assignment date was off by a few months. Most of the affected records were over five

years old and the HWC does not require more precision for its business process.

Another duplication problem in the assignment history table was that there were

duplicate history entries for SMs who were promoted after they moved into MFH. The

HWC does not normally know when a SM is promoted, however, it appears that

whenever the HWC updated the SM's record in FAMIS, the application automatically

added a new record to the history table, even though the SM did not move to another

MFH.

As with the SERVICEMEMBER table, we queried the ASSIGNMENT TABLE

to find duplicate records, then deleted the duplicate record in the pair with the least

amount of information about the assignment.

5. FAMILYMEMBER Table

We found two source tables for the FAMILYMEMBER data migration: one for dependents currently living in MFH, and one for past occupants. The family member historical table in FAMIS (ASDPS) contained over 10,300 records. Of those, almost 1200 were duplicate records. Normally, the key attributes for this table would be SSN and RelationCode. Because many of the columns, such as BirthDate and RelationCode, had null values, we could not designate a key for the table. We decided to create an auto number column as the key and use the SSN to link the family members to service members. We were able to import the table for current occupants using the same process; however, after importing the table we were able to create a more meaningful key using the SSN, FirstName, and BirthDate columns (which were not null in this table).

6. FLOORPLAN Table

After reviewing this table, we found another example of multiple naming conventions. The unit designation field was defined in several ways, depending on the table. For example, the FAMIS table tracked the designation for a 3-bedroom floor plan, allocated for field grade officers, as either "3BD FG-LM" or "3BD FG-PA", depending on the base in which the unit is located. Several of the HWC Access tables defined the units as either "3FG" or "FG-3BD-2BT", depending on the table (and the base). In order to get the same designation for each unit, we queried the FLOORPLAN table and joined it with the MFHUNIT table as shown in Figure 5-5. Then, we created a new MFHUNIT table with a standard naming convention for the Designation column.
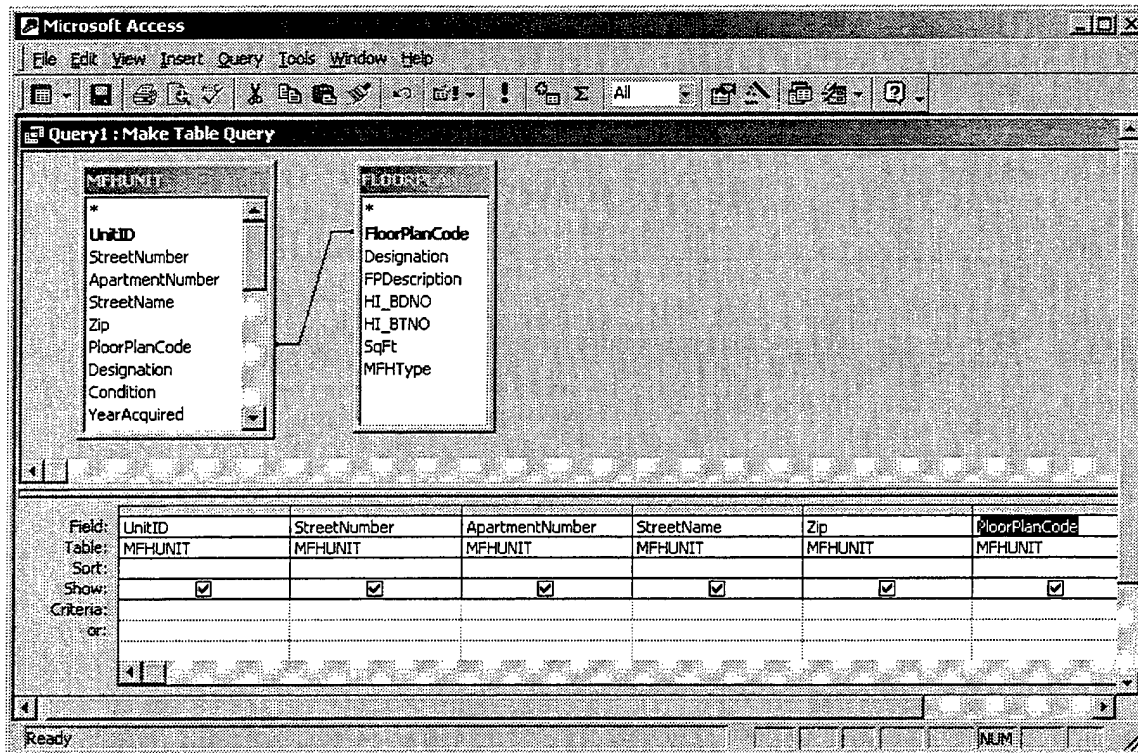
Figure 5-5.   Updating the Designation Column

7.   APPLIANCE Table

The HWC uses two Access tables to track appliances: one for OMC and POM, and one for NPS and La Mesa. The Access tables for tracking appliances are flat files— they don't relate to an appliance entity. If you delete an MFH record in the table, the appliances are also deleted.   According to the HWC, every MFH should have government-owned appliances, such as refrigerators, dishwashers, and stoves. Given that the HWC manages about 2,200 units, they should maintain information on approximately 6,600 appliances. However, the HWC Access tables only contain 1,150 appliances.

55

FAC-REC	Res Add	STOVE TYPE	SERL.	REF TYPE	SER.	DISHWASHER TYPE	SER
1	370 A BERGIN DRIVE	SUNRAY	9306136020	WHIRLPOOL	SX1728763	HOTPOINT	U2601649
2	370 B BERGIN DRIVE						
3	370 C BERGIN DRIVE						
4	370 D BERGIN DRIVE	SUNRAY	9306136031	GIBSON	0348602292	HOTPOINT	UX601670X
5	370 E BERGIN DRIVE	SUNRAY	9110158970			HOTPOINT	U2601669X

Table 5-7.   Original Appliance Tracking Table for La Mesa, LA-APPL

First, we exported the Appliance tables (APP-POMA and LA-APPL) to Excel, copied the UnitID column in front of each appliance type column, and linked the unit with the appliances. Next, we added columns to specify the type of appliance. Then, we cut and pasted the information in order to align the appliances into unique rows, as shown in Table 5-8.

UnitID	TypeOfAppliance	Manufacturer	SerialNumber
1	S	SUNRAY	9306136020
2	S		
3	S		
4	S	SUNRAY	9306136031
5	S	SUNRAY	9110158970

Table 5-8.   Creating Unique Appliances

Still in Excel, we added a unique number to each appliance, resulting in Table 5-9, which we then exported to the APPLIANCE table in our Access database.

ApplianceID	UnitID	TypeOfAppliance	Manufacturer	SerialNumber
1	1	S	SUNRAY	9306136020
2	2	S		
3	3	S		
4	4	S	SUNRAY	9306136031
...	...	...	...	...
576	1	R	WHIRLPOOL	SX1728763
577	2	R	WHIRLPOOL	SX1728991
578	3	R	WHIRLPOOL	SX1728976
579	4	R	GIBSON	0348602292
...	...	...	...	...
1151	1	D	HOTPOINT	U2601649
1152	2	D	HOTPOINT	U2601111
1153	3	D	HOTPOINT	U2601102
1154	4	D	HOTPOINT	UX601670X

Table 5-9.   Summary of the Completed APPLIANCE Table

Finally, we used the query in Figure 5-6 to weed out null records, such as the second and third records from Table 5-9 (above), and create our APPLIANCE table.
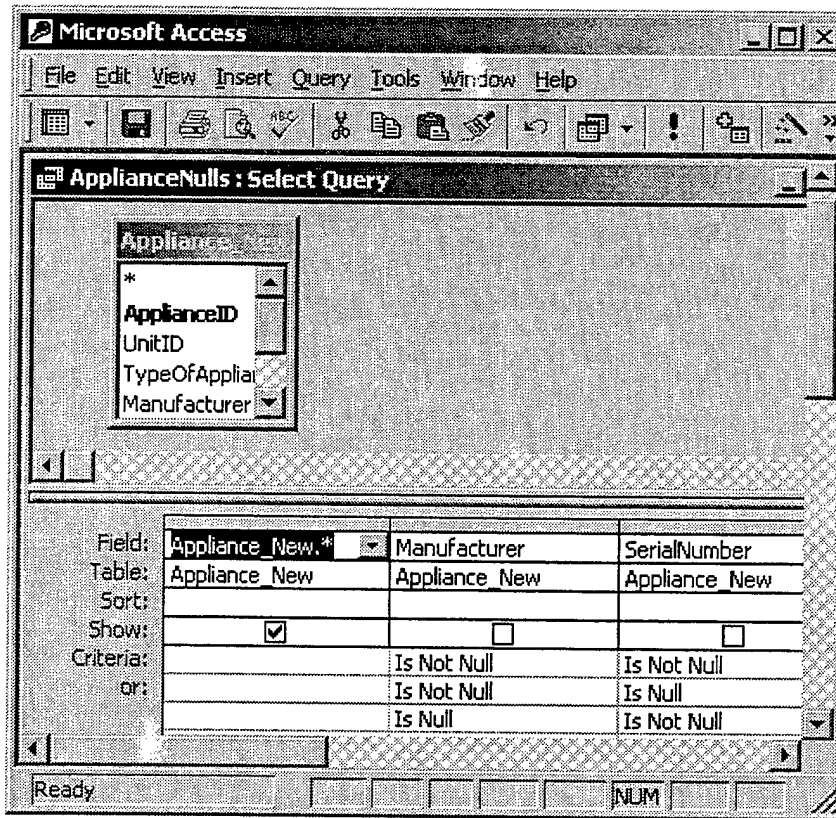
Figure 5-6.   Access Query to Weed Out Null Appliance Rows

8.   PRIVATERENTALPROPERTY Table

Some of the address information, such as city and zip, were missing from a significant number of records. Initially, we planned to place only the zip in the table, linking it to our ZIPCODE table, which contains the city and state for each unique zip. However, for our prototype, we decided to leave the fields and migrate them "as is" for the referral portion.

In this table, we also noticed data entry errors. One example is in the RF_DDEL column, which is the date to delete the referral listing. An unknown system user at the HWC entered "2270" for one of the referral properties, meaning that the owner wishes to remove the referral listing 269 years from now! We assumed that this was either a data

entry error, or the owner had great long-term property management skills. We can only guess his/her intention. In order to truly clean this kind of error in the table, someone at the HWC would need to verify the information through other sources.

C. SUMMARY

We began the data migration process with a physical database design and over 80 separate operational data files from the HWC. In order to migrate the operational data into our database, we had to perform thorough, sometimes tedious, data cleansing to find the desired data. While finding the correct data and understanding the existing schema was difficult enough, we needed a data crosswalk in order to map out the data's migration path. Not only did we extract data from FAMIS, but we also had to extract and contour the flat files found in the HWC's Access database.

We encountered many data quality issues as we migrated the data. Normally just harmless mistakes, misspelled words created relational roadblocks in our tables. Other data issues, such as inconsistent naming conventions and null column values, tested our querying abilities and forced us to learn the schema inside out. In the end, however, we successfully converted the data into our prototype, allowing us to move on and create the user interface.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. PROTOTYPE IMPLEMENTATION

A. INTRODUCTION

Database tables by themselves are boring and non-intuitive as a basis for users to perform business tasks. The tables alone cannot capture the business process and enforce business rules. In order to perform efficient operations on multiple records in tables, users need to know a query language, such as the standard structured query language (SQL). However, for the HWC, as is the case for an ordinary business, we do not require each user to possess SQL skills—that would create an unnecessary training burden. Instead, we take advantage of numerous database applications available in the software market. Database applications, such as Access, provide five major functions:

- Create, read, update, and delete (CRUD) views;
- Materialize or format views;
- Enforce constraints;
- Provide security and control mechanisms;
- Execute business logic (Kroenke 267).

It came as no surprise that potential users of our system expected an easy to use graphical user interface (GUI), whether through the database application, or an Internet browser. Therefore, we needed to create a series of layered menus in Access, known as switchboards, as the primary front-end to the database. Beneath each menu, we also needed to create the relevant forms, reports, and other functions provided by a database application to satisfy the requirements identified in the use cases.

Because one of our use cases required implementing Internet-based housing applications, we also needed to design the Web-based GUI and provide connectivity to the HATS database.

61

B. FORMS

Since our system requirements identified Access as the users' interface with the database, we created menu views to allow point and click navigation. In Access, a switchboard is the equivalent of a menu. It is just another form that allows us to place text, pictures, and buttons on the screen to facilitate user interaction with the system. The main switchboard, shown in Figure 6-1, displays several command buttons, each labeled after the functions identified in our functional decomposition.
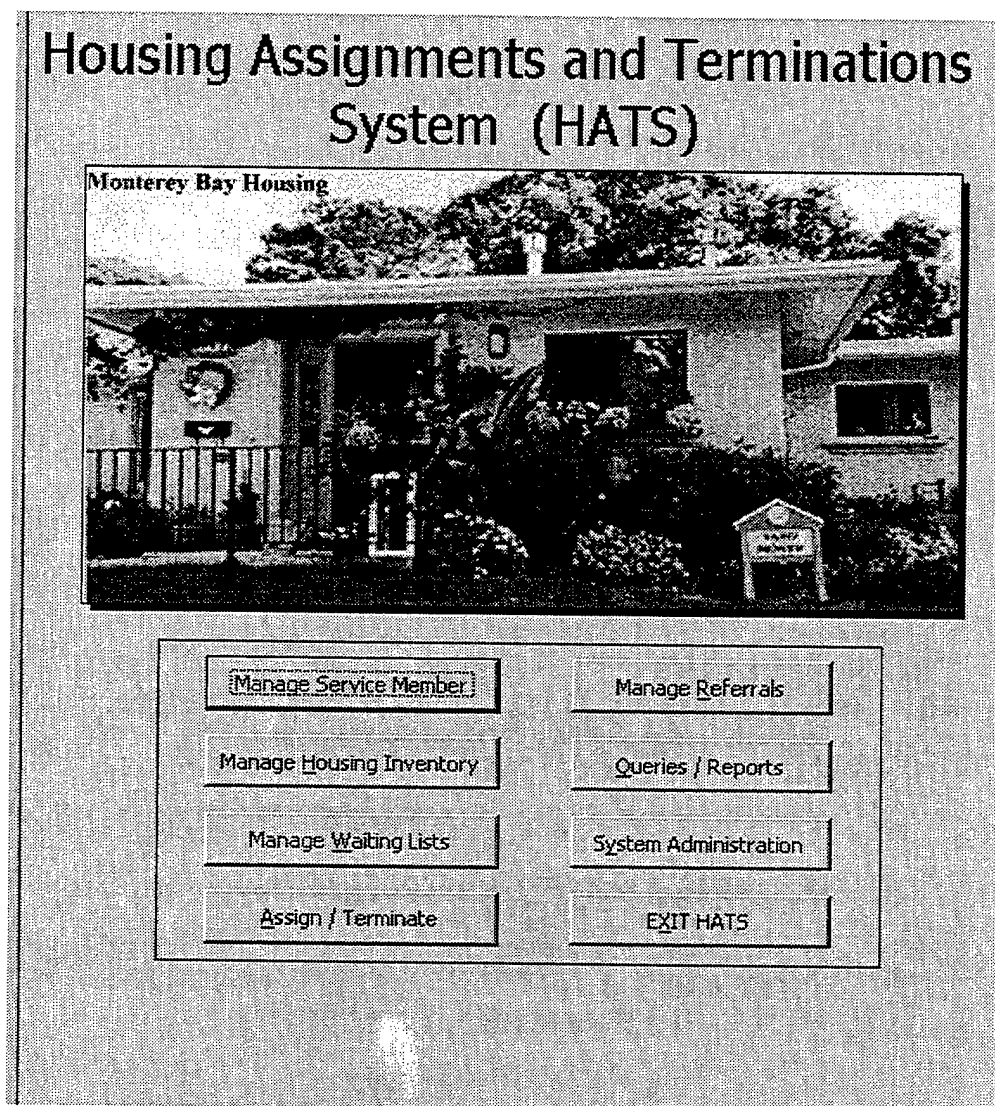
Figure 6-1. HATS Main Menu (Switchboard Form)

Each function's command button links the user to the next level of functions. For example, the main switchboard acts as the main menu for HATS. The "Assignments and Terminations" button on the main form links the user to the "Assignments and Terminations" functions in the level-2 functional decomposition. From that menu, each button links the users to forms, reports, and other functions specified in the use cases.

We returned to the use cases in Appendix A and analyzed them to determine what forms the users of our system may need to interact with the database. Table 6-1 lists the name and purpose of each form and traces the requirement back to the use cases.

We used tools in Access to create our forms and enforce the HWC's business rules. Almost all of our forms were designed to use drop down menus, check boxes, or other controls to limit a user's input to match valid domains. For example, in order to edit a service member's record using the edit SM form in Figure 6-2, the user must select an SSN from a drop down menu, which forces the user to input a valid SSN. Otherwise, the user could type an incorrect SSN and the application would display an error message, indicating that no such record exists.

Form Name	Form Purpose	Use Case
DD Form 1746	Application for Housing	UC 10
Add SM to List	Add SM to a Waiting List	UC 25
Edit SM	Edit SM Information	UC 15
New CRL	Enter CRL Search Criteria	UC 60
Edit Wait Status	Edit SM Wait Status	UC 35
Edit Freeze	Edit Freeze Zone	UC 40
Match SM MFH	Match SM with an MFH	UC 45
Offer MFH	Offer MFH to SM	UC 45
Assign	Make an Assignment	UC 50
Intent to Vacate	File Intent to Vacate and Schedule Clearance Inspections	UC 55
Terminate	Terminate Assignment	UC 55
Update MFH	Update MFH Attributes	UC 65, 85
Add MFH	Add MFH to the System	UC 65
Delete MFH	Delete MFH from the System	UC 75
Add PR Property	Add Private Rental Property	UC 70
Update PR Property	Update Private Rental Property	UC 70
Delete PR Property	Delete Private Rental Property	UC 75
Add Owner	Add Property Owner Information	UC 80
Update Owner	Update Owner Information	UC 80
Delete Owner	Delete Property Owner from System	UC 80

Table 6-1.   Forms Required According to Use Cases

Figure 6-2.   Using Drop-down Menus on Forms

C.   PROVIDING INTERNET CONNECTIVITY TO THE DATABASE

Since one of the main goals of our prototype was to demonstrate online application processing, we created a test HWC Web page and linked it to our database using common Web applications and technologies: Web serving and development applications, Active Server Pages (ASP), and Open Database Connectivity (ODBC).

Active Server Page (ASP) is a Microsoft Internet Information Server (IIS) feature that supports scripting languages to interact with applications, such as a DBMS.   The ASP code is part of the source code on the Web page, which is written in Hyptertext Mark-up Language (HTML).   It is the ASP code that carries out the instructions to insert the data from the Internet application form into the HATS database tables.

Open Database Connectivity (ODBC) is a standard application program interface (API) for connecting application programs to databases (Kroenke, 2000, p. 342).   By using ODBC, we can link our Web site (application) to our database through the connections shown in Figure 6-3.   One advantage to using the ODBC standard is that if someone decides to move HATS to another DBMS, such as Oracle, the HWC would only need to configure another ODBC connection.   They would not need to make any changes in either the Web site or the database.   Since we used the Access DBMS, we configured the Web host server's ODBC connection to allow access to the data source (our prototype database) using an Access DBMS driver.
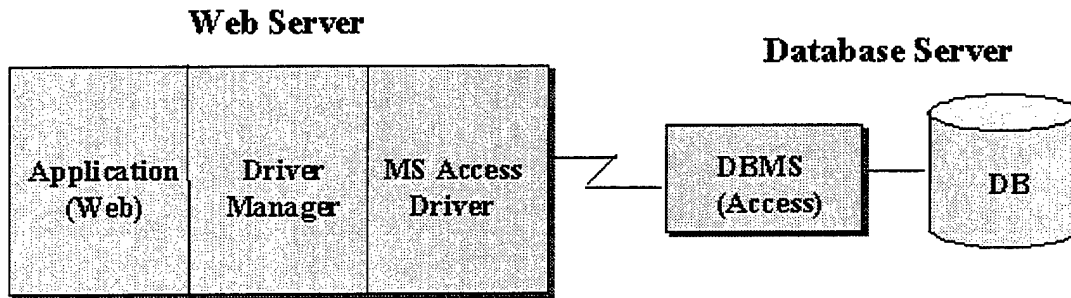
Figure 6-3.   ODBC Architecture in Relation to HATS (After Kroenke, 2000, p. 343)

Before we connecting our prototype to the world, including some Internet browsers with malicious aspirations, we decided to implement a minor safeguard to protect the integrity of one of the main tables (SERVICEMEMBER). We created a staging table (shown in Figure 6-4), APPLICANT, to record information about service members applying for housing. When service members submit their personal information to the HWC through the ASP site, the data is stored in the APPLICANT table. Otherwise, anyone could enter data into the SERVICEMEMBER table and the HWC would have to screen the table for data entry anomalies. By creating the APPLICANT table, the HWC can screen the information, then without re-typing data, click a macro button and transfer the information to the SERVICEMEMBER table.

Using Microsoft FrontPage 2000, we created a simplified version of the HWC's housing application (DD Form 1746). FrontPage has Web form creation tools that make the process as easy as pointing and clicking. In addition, FrontPage created the ASP commands, so we did not have to learn another sub-language for this task. In order to complete the other end of the ODBC loop, we opened the form properties in FrontPage and routed the results of the application to our database.

Figure 6-4.   APPLICANT Table for Recording Online Applications

Finally, we published our housing application (DD Form 1746), shown in Figure 6-5, on a Microsoft Internet Information Server (version 4). We tested the connection by entering fictitious applications through a Web browser, then confirming that the applications were successfully added to the APPLICANT table in our prototype.

MONTEREY BAY
MILITARY HOUSING - ON LINE!
APPLICATION FOR ASSIGNMENT TO HOUSING

SSN ____-__-____   Paygrade   03

Last Name [_____]   Branch of Service [Navy ▾]

First Name [_____]   Report Date   [mm/dd/yyyy]

MI [____]

Address [_____] ▲▼   (Street, City, State, Zip)

Email [_____]   Marital Status   [Married ▾]

Daytime Phone [_____]   Separated From Dependents? [No ▾]

Evening Phone [_____]   Request Housing For   [Self and Family ▾]

Assignment (in Monterey) [Student NPGS ▾]   Total Bedrooms Authorized [2 ▾]

Departing Command [_____]

Remarks [_____] ▲▼

[Submit] [Reset]

Figure 6-5.   Summarized Version of the Housing Online Application (DD Form 1746)

D.   REPORTS

Just as we determined which forms we needed to create, we returned to the requirements analysis phase of our prototype to identify the required reports. The "View and Print Reports" use case, UC 90, stated that users need several reports, which we identified in our functional decomposition and summarized in Table 6-2.

Reports	Frequency
Wait Lists	Weekly
Housing Statistics	Monthly
Coast Guard	Monthly
Mayors'	Monthly
Vacate	Monthly
Occupants	Monthly
Non-eligible	Quarterly
Utilization	Annually
School Children	As Needed
Assignment History	As Needed

Table 6-2.   Reports Derived From Functional Decomposition (and UC 90)

Figure 6-6 shows a segment of the monthly occupant report, which is organized by neighborhood, then by address. We created this report from a query that searches for all current assignments. Using the same process, we created the remainder of the reports listed in Table 6-2. These are shown in Appendix D.

70

Occupant Report

Neighborhood	FITCH						
Street #	Street	Apt.	Unit #	Last Name	First	Paygrade	Dependents
300	AACHEN ROAD		3289	WILLIAMS	CLIFTON	O3	4
301	AACHEN ROAD		3291	ANDERSON	TIM	O3	3
302	AACHEN ROAD		3292	TAYLOR	TARL	O4	5
303	AACHEN ROAD		3293	VEGTER	HENRY	O2	3
304	AACHEN ROAD		3294	GORAK	MARK	O3	4
305	AACHEN ROAD		3295	RULOF	ROBERT	O2	2
306	AACHEN ROAD		3297	DOLLOFF	KATHERINE	O3	2
307	AACHEN ROAD		3298	DEVINE	CRAIG	O3	5
308	AACHEN ROAD		3299	KUBISTA	THEODORE	O3	2
309	AACHEN ROAD		3300	DUGONE	THEODORE	O3	4
310	AACHEN ROAD		3301	DIMMICK	MATTHEW	O3	3
311	AACHEN ROAD		3302	SPENCE	EDWAR	O3	3

Figure 6-6.   Occupant Report

E.   SUMMARY

After creating database tables, we implemented the features of a database application (Access) to provide the reports, forms, views, and other functions necessary to support the HWC's business process. We created a GUI within Access to facilitate easy navigation between the system's functions.

Because the system requires Internet-based input, we created a GUI for Internet users and confirmed that information entered via this GUI was inserted into our database. By using common Web design and publication applications, we were able to create a simple housing application form on the Internet and establish a link to our database. Although a prototype, the Web page successfully inserts service member applications to the HWC, eliminating the need for the HWC to re-type information.

71

As with any prototype, our system can benefit from several iterations of user feedback before going "full up". As much as we attempted to make the GUI useful and intuitive, we cannot expect to capture their needs on the first attempt. In addition, since we performed an intensive data migration effort, any plans to develop a "full up" system should include a data verification procedure before going live.

For an operational system we recommend migrating to more powerful applications, such as Dreamweaver for Web design and Oracle for the database. Although FrontPage is easy to use, it is not as robust as Dreamweaver, which is preferred by the majority of professional Web designers. As for the database application, Oracle's database is far superior in terms of performance, but more expensive than Access.

We used Access in the prototype for two main reasons. First, we were familiar with the development GUI and convenient interfaces with other Microsoft applications, such as FrontPage, Word, and Excel. Second, the HWC already owned the software license for Access, so there would have no additional expense for implementing the prototype.

Ideally, the front-end to the system should eventually reside on the Web. By using an Internet browser as the GUI to the database, the HWC users and its customers can interact with the system using a wide range of Internet browsing devices, including personal data assistants (PDAs). Additionally, the HWC would not need to purchase and install Access on their data entry devices.

VII. DECISION SUPPORT TOOLS

A. INTRODUCTION

Up to this point, we have created a relational database with a user-friendly GUI to input and edit data and view text reports. What we have not done yet is provide tools for users to manipulate, organize, and analyze data and information in order to make better decisions. A decision support tool (DST) is any tool that enhances the ability of humans to make decisions. Our goal is to provide tools that transform data and information into knowledge (illustrated in Figure 7-1), which HWC managers, counselors, and customers can use to make better housing utilization decisions.
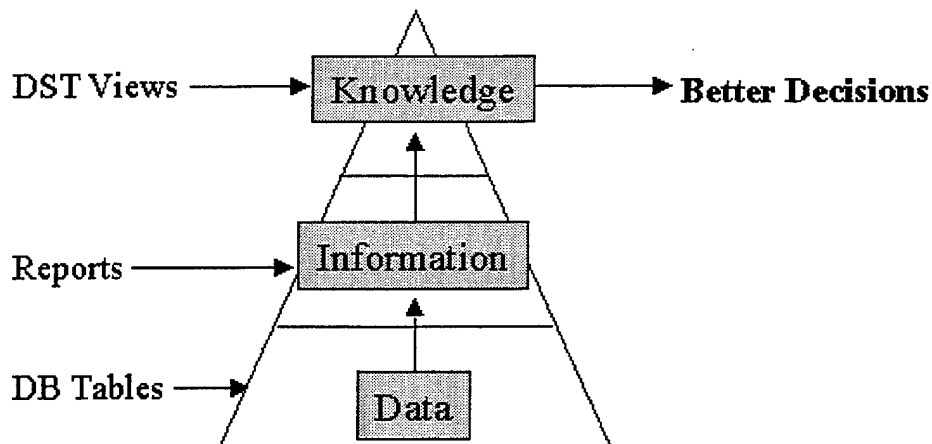
Figure 7-1.   DSTs Foster Knowledge and Better Decisions

Customers and the HWC will have access to data and information through the DBMS to formulate knowledge and make decisions. However, because they are human, they are limited in their capacity to process data and information into knowledge. We will provide DSTs for both the decision makers at the HWC—the HMSs and the supervisors—and their customers, the service member deciding which MFH s/he wants to

73

live in. A poor decision about which houses to offer (by the HWC), or which house to occupy (by the service member), can have a serious impact on quality of life and job satisfaction over the duration of the assignment, possibly leading to negative impacts on retaining service members in the future.

Often, robust DSTs are part of a larger, sometimes expensive decision support system (DSS). For our system, however, we considered inexpensive, readily available applications that provide useful DST functionality. Since we are already using Access as our DBMS, we decided to use some of the add-on tools provided with Access. The other DST that we chose to use is a geographic information system (GIS). Because many Web sites today are using GIS technology, we decided to implement a tool that is gaining in popularity and acceptance in homes and businesses around the world. Additionally, GIS provides an intuitive, visual display of housing locations which can assist the decision makers and customers in the housing assignment process.
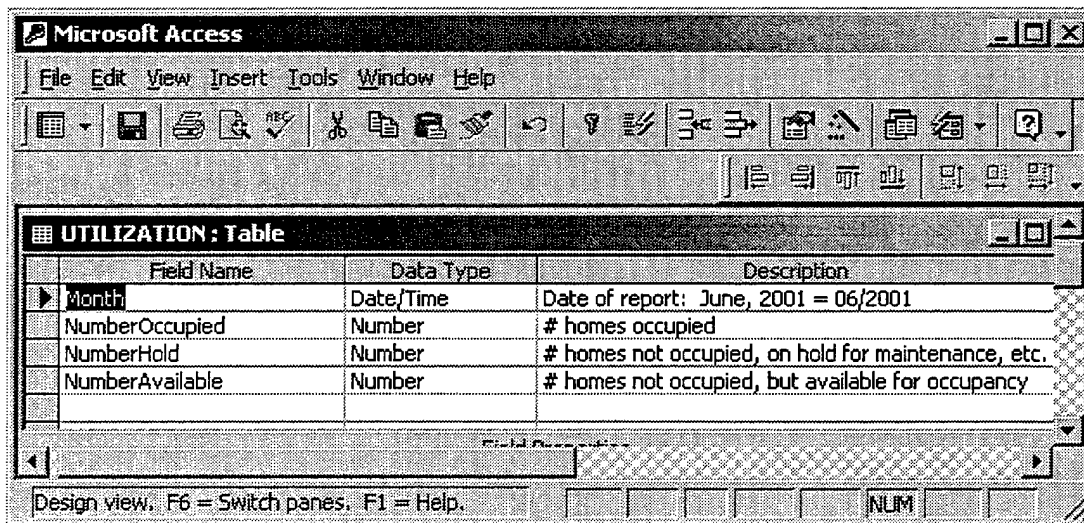
B.   MICROSOFT ACCESS TOOLS

Since we have created the database application in Access, it is only natural to take advantage of the convenience of the embedded tools in the DBMS. Access provides built-in functions that enhance a person's ability to make good decisions. Some DSTs provided by Access include:

- Graphs;
- Charts;
- Calculations;
- Filters;
- Queries;
- Macros.

74

Some of the DSTs provide visual information, such as graphs, while others provide filtered or manipulated views of existing data, such as queries. The real power of Access is the ability to combine tools. For example, users can query the database (or another query), perform calculations, filter the information, and display the results in a graph or chart. Using these embedded tools, we can improve decisions about such concerns as housing utilization, assignment waiting queues, and other critical success factors determined by the HWC.

In the following example, we show how to combine tools in Access to create views that can enhance a user's ability to make better decisions. First, we create a new table (shown in Figure 7-2) that keeps a record of utilizations statistics, such as the number of occupied and unoccupied houses. The key of the table is Month, which is the month of the report and the year. This table allows the HWC to keep a historical utilization file and analyze trends that could help predict future rates.

Figure 7-2. UTILIZATION Statistics Table

Next, we create queries to generate the statistics, such as the number of occupied houses, and store the data in the UTILIZATION table. Figure 7-3 illustrates the query, as seen in the query by example (QBE) view in Access.
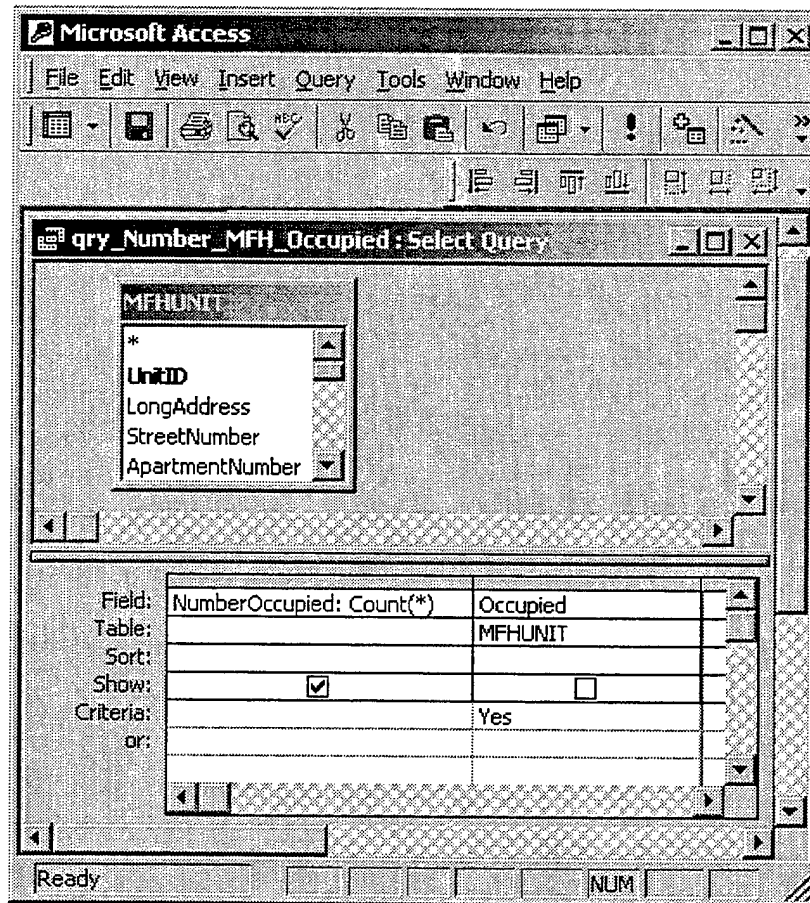
Figure 7-4.   Access Query to Count the Number of Occupied Houses

Finally, we create a chart of the utilization statistics by selecting the UTILIZATION table and clicking on the menu choices, "Tools, Office Links, Analyze it with Excel". This tool loads the table data into an Excel spreadsheet, where we use the chart wizard function to quickly create a chart depicting the historical utilization. The chart, shown in Figure 7-5, helps the HWC visualize utilization patterns that may not be evident when viewed in tables or reports.

The HWC can view the utilization chart by selecting the "View Utilization Statistics" button on the "Queries and Reports" main menu. The button contains a macro that runs the queries, updates the UTILIZATION table, and opens the chart shown in Figure 7-5. This chart could help the HWC recognize that there is a decreasing trend in the number of occupied units, and an increasing trend in the number of available units. This revelation may trigger the HWC to make decisions concerning renovations, or other utilization decisions that take advantage of this knowledge.
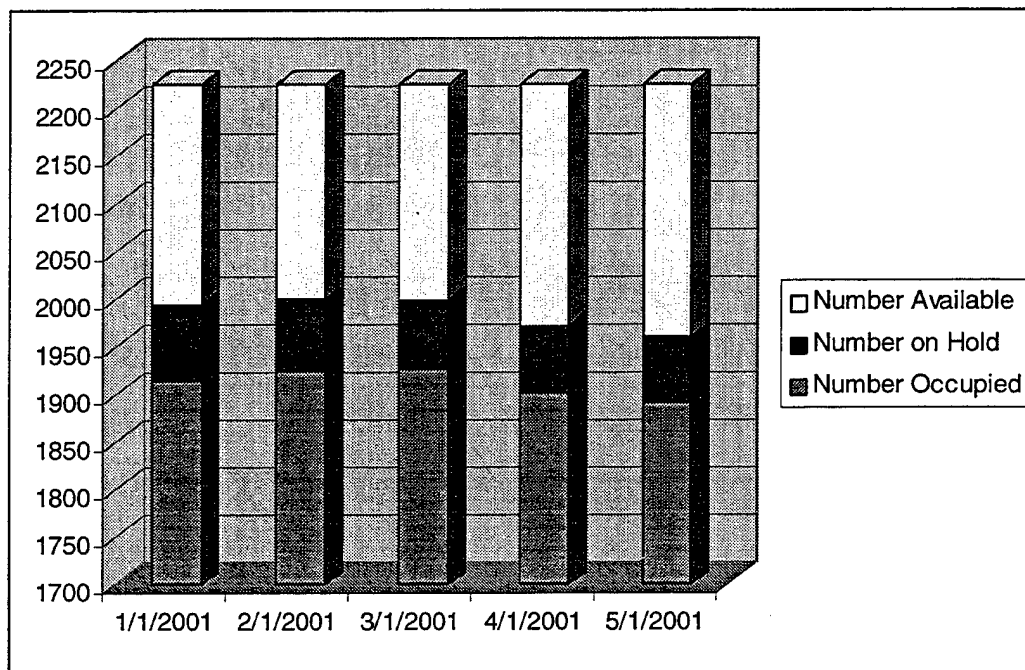
Figure 7-5.   Utilization Chart

C.   GEOGRAPHIC INFORMATION SYSTEM

So far, our prototype displays data in forms, datasheets, charts, and reports. Since the focus of our system is assigning people to houses, we have yet to deal with the spatial aspect of the assignment and termination process. Customers often want to know where prospective houses are located, or how close they are to schools, churches, commissaries,

playgrounds, and other families. According to Environmental Systems Research Inc., 80 percent of all data has a spatial component (2001). A picture is truly worth a thousand words, or in this case, several thousand words. Traditional information systems fail to capture and display the significance of spatial information. Geographic Information Systems (GIS) provide the ability to present spatial data in a meaningful way to provide information that may lead to knowledge and better decisions.

A GIS is a computer system for "capturing, storing, checking, integrating, manipulating, and displaying data using digitized maps" (Aronson, 1998, p. 243). A GIS is more than just a map—it displays layers of spatial data that can be captured (or printed) in a map, if desired. In addition, a GIS maintains tables of data that can be queried, manipulated, and displayed with other layers for analysis.

By integrating our DBMS with a GIS, we can make a more robust system that displays visual information that could otherwise be overlooked by decision-makers. Without the GIS, users must interpret data from database tables, reports, and static street maps. A GIS puts the separate layers of data (as illustrated in Figure 7-6) into perspective, creating dynamic views of the business environment.

Data from many sources can be combined to create individual layers, that when combined, create a meaningful view of information. Figure 7-7 illustrates the fact that a GIS combines data from several sources to create a synergistic and robust view of the research environment.

We chose ESRI's ArcView (version 3.2) as our GIS application for this prototype primarily due to its available and wide acceptance by business and government agencies. ArcView allows users to retrieve data directly from the database, which in our case is the

HATS database in Access. In order to retrieve the data, we create an ODBC connection

with our Access database, as illustrated in Figure 7-8. While connected to our database,

we query the MFHUNIT table and create new tables in our ArcView project for all

houses, including currently available houses and those available in the near future.
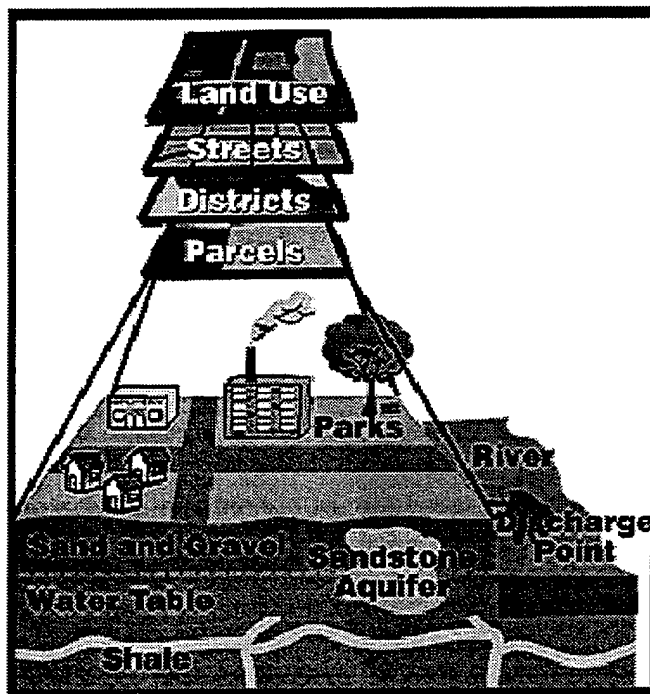
Figure 7-6.　　GIS Layers of Data (From ESRI)
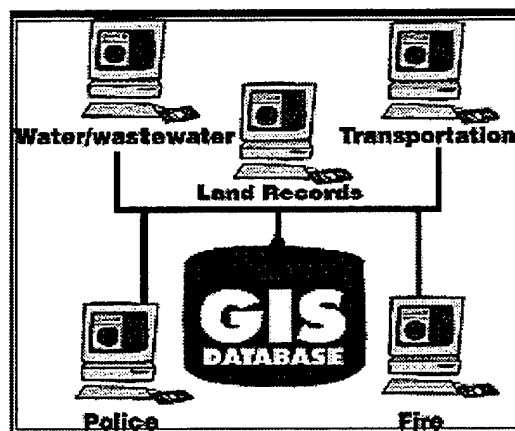
Figure 7-7.　　Connecting Data Sources to a GIS Database (From ESRI)
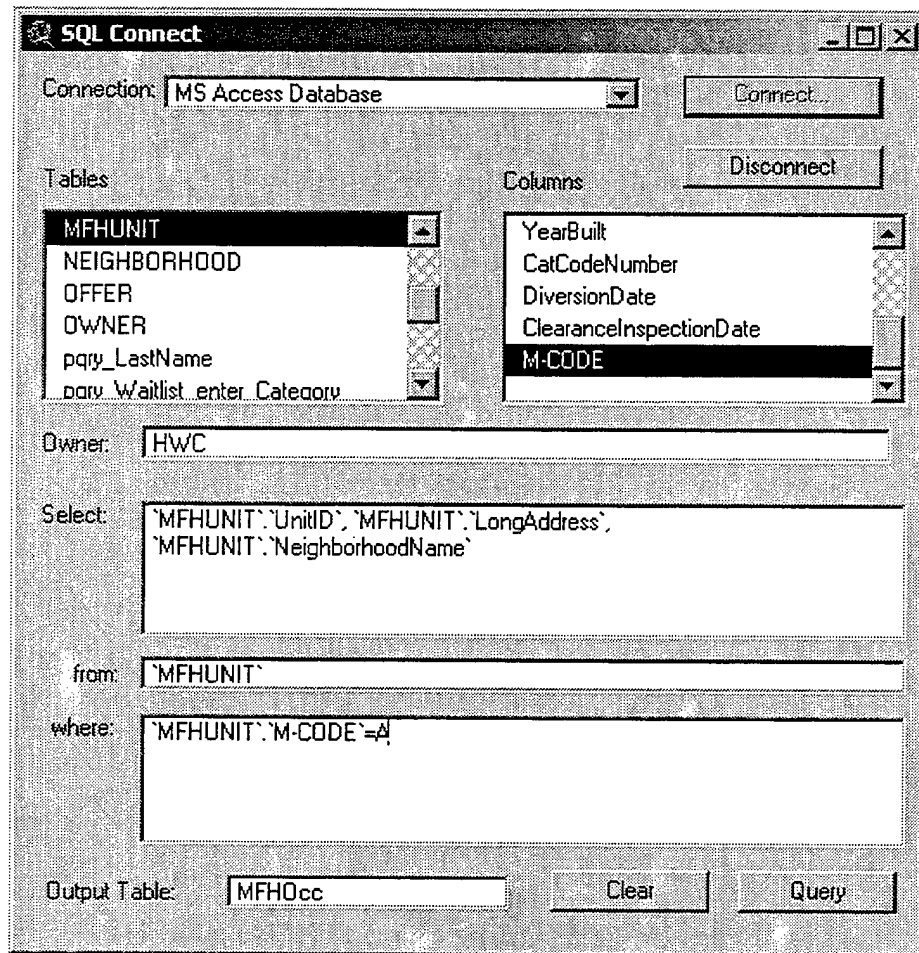
Figure 7-8. Querying the Database for Available Units and Adding the Results as a
New Table in ArcView (Also Using ODBC)

Before we can view our houses in a spatial manner, we need to add a reference

layer containing spatial data and match our addresses with the layer. In other words, we

need a reference map with coordinate information that we can overlay with address icons.

ArcView shapefiles are data files containing such reference information, which is

technically referred to as a geocoding index. First, we use a shapefile from ESRI (with a

geocoding index for our houses) as our bottom data theme. Next, we geocode our MFH

addresses by linking the address fields (street number, street name, and five digit Zip

code) in the tables to the geocoding index in the reference shapefile. As we match each

table from our database, such as the MFHUNIT table with all houses in the HWC inventory, ArcView creates a new theme for each. Figure 7-9 shows the reference theme, named Street.shp, and four themes for housing status: available now, available in less than 30 days, available in more than 30 days but less than 60 days, and all units. In Figure 7-9, the Street theme and the All Units themes are selected (as indicated by the check mark in front of the theme name) and displayed in the viewing window. Users can change theme selection to capture a number of different scenarios. For example, we change views so that a user can display just the houses currently available for occupancy, shown in Figure 7-10.

For our system, users need to drill down and see more information about the themes. For example, they may want to know what category of housing is available and where they are located. ArcView provides a query function to select records from a theme based on criteria, such as housing category, and highlight the query results in the theme view. For example, we decide to specify a housing category in the Available Now theme. After making the theme active, we view the theme table containing all of the records for the available houses. ArcView provided a query window, shown in Figure 7-11, to facilitate our query. We select the category 3JOOMC (three bedroom, junior officer, at Ord Military Community). Notice that ArcView user interface helps the user select the values for the query by displaying all of the valid values for the category field. After making our selection, ArcView displays the entire theme table and highlights the matching records, as shown in Figure 7-12. More importantly, ArcView also updates
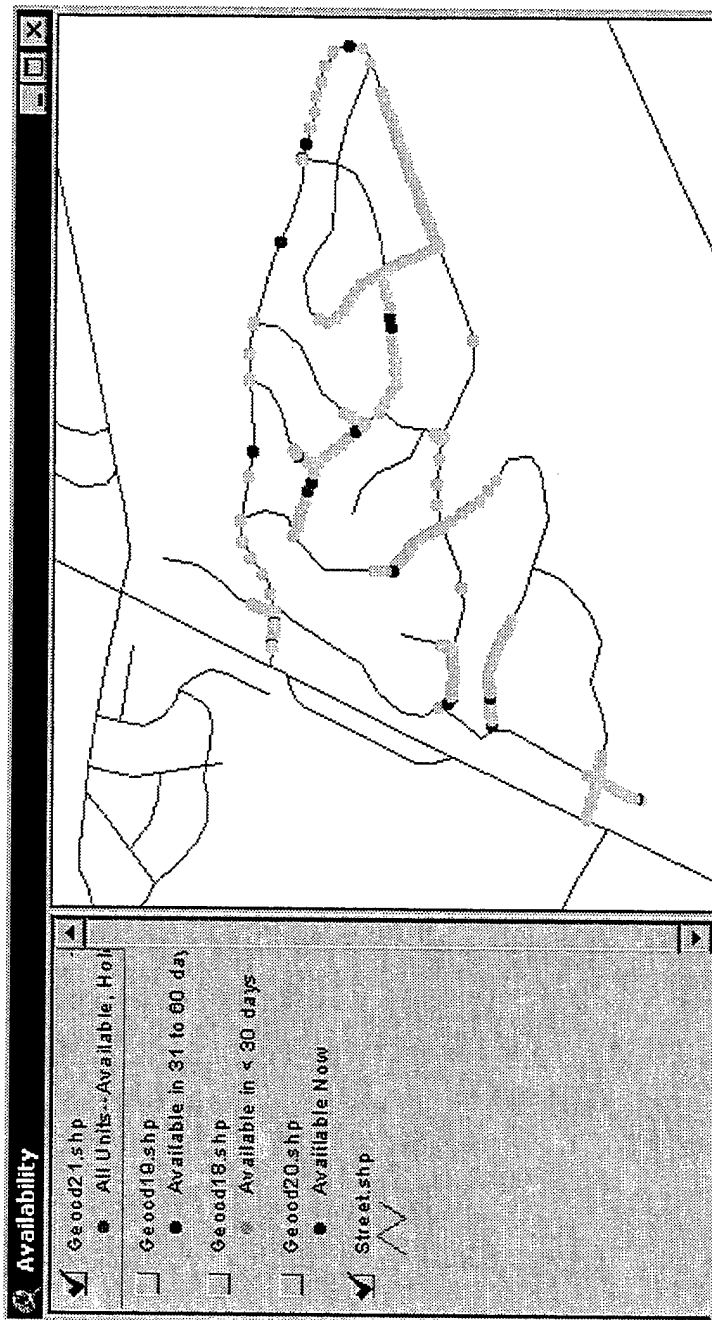
Figure 7-9.   View of All Units Theme with Occupied Units Highlighted
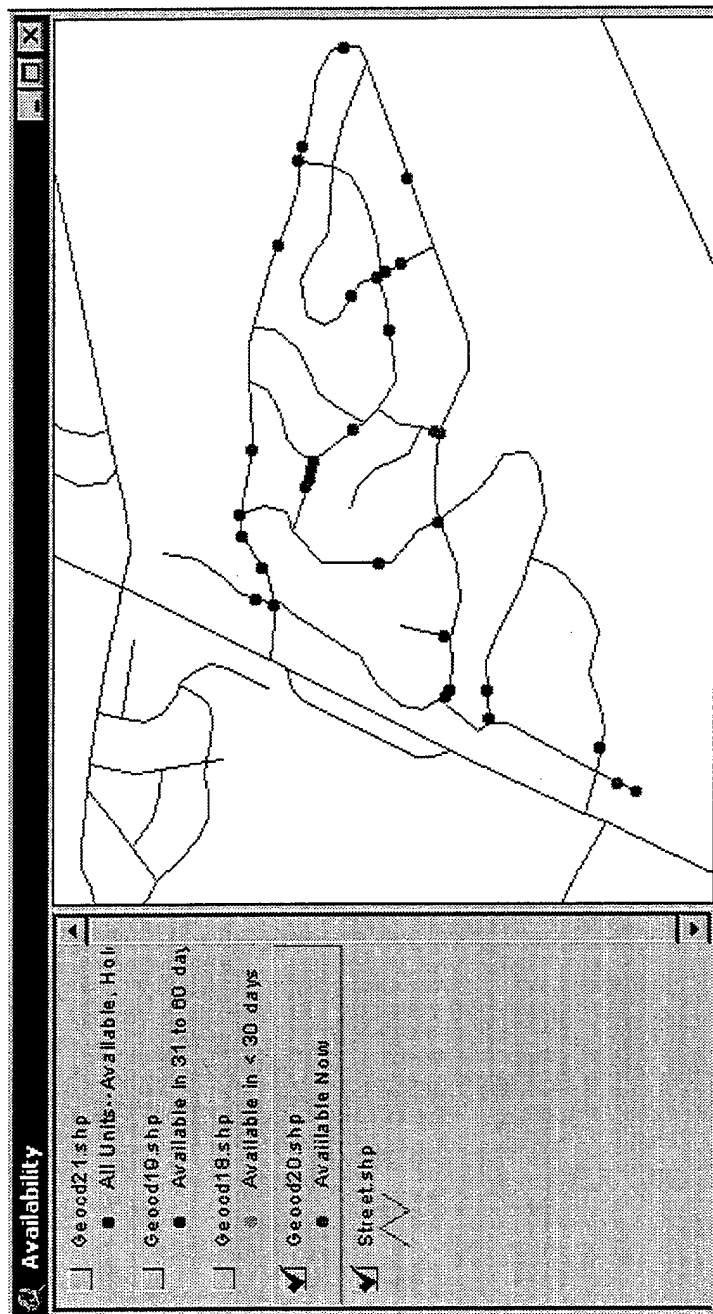
Figure 7-10.   View of Available Units

the theme view, shown in Figure 7-13, to give the user a more useful visual of the query results.
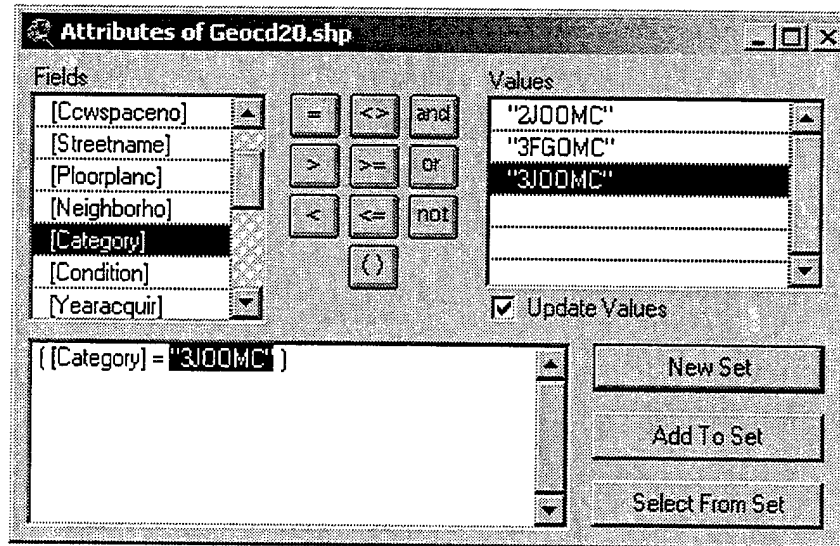
Figure 7-11.   Selecting a Housing Category to View

Figure 7-12.   Result of Query for Category "3JOOMC" Houses

84
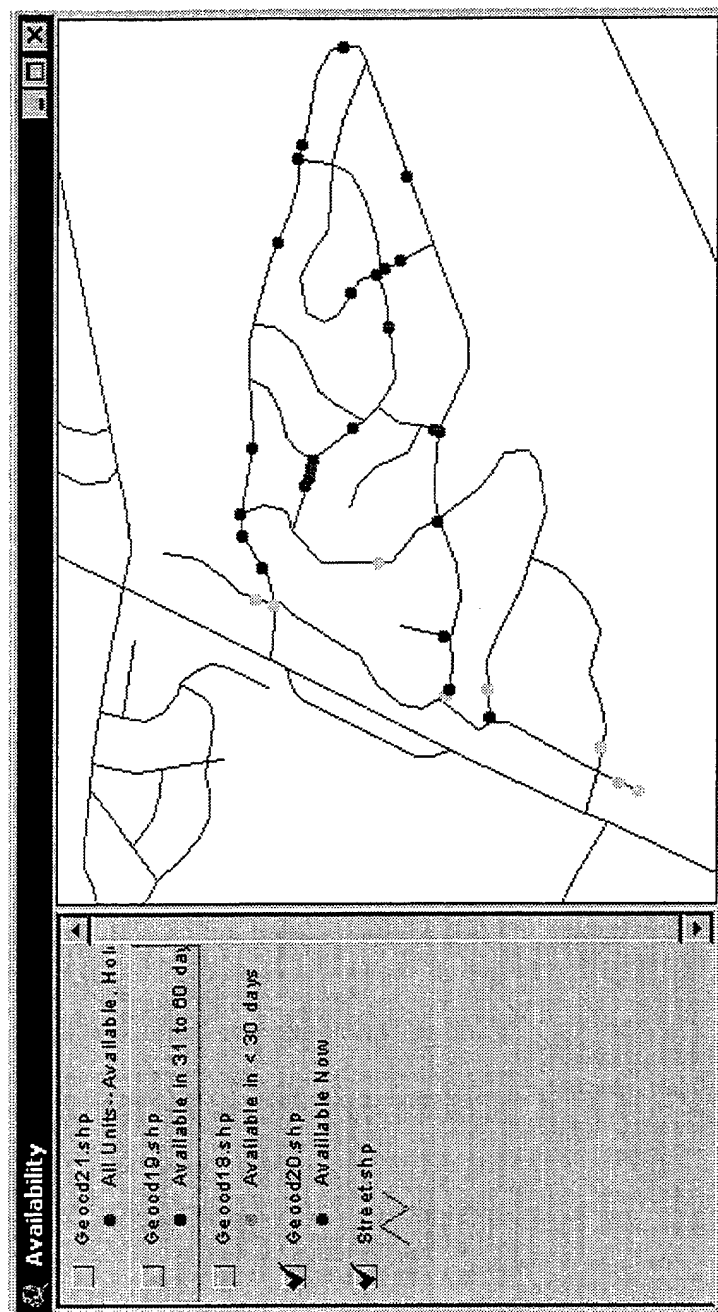
Figure 7-13.   Map Showing Category "3JOOMC" Available Units

Based on our experiences, one of the most important criteria for housing selection is the demography of the area. Service members and family members want to know the number and characteristics of the people, in relation to their age, for example. More specifically, parents are concerned about the ages of the children in the area because they want their own children to have playmates close to their house. Because our prototype HATS database contains a FAMILYMEMBER.Birthdate field, we can extract the ages of children from our database and create demographic overlays to view in conjunction with available houses. Using a simple query, we create demography themes for children in several age categories:

- Newborn through 1;
- 2 through 4;
- 5 through 6;
- 7 through 9;
- 10 through 13;
- 14 through 18.

By combining these demography themes in a view with the available houses and the street themes, we create a tool that allows customers to find available houses in areas matching their preferences. For example, when a customer wants to find an available unit surrounded by 7-9 year olds, s/he selects the appropriately labeled theme, along with the Available MFH and Street themes. The resulting graphic, shown in Figure 7-14, provides the customer visual information that could improve the assignment decision and increase customer satisfaction. Without this tool, the customer could only discover the ages of children in the area of the available houses by a physical inspection of the neighborhood, or by viewing a printed text report.
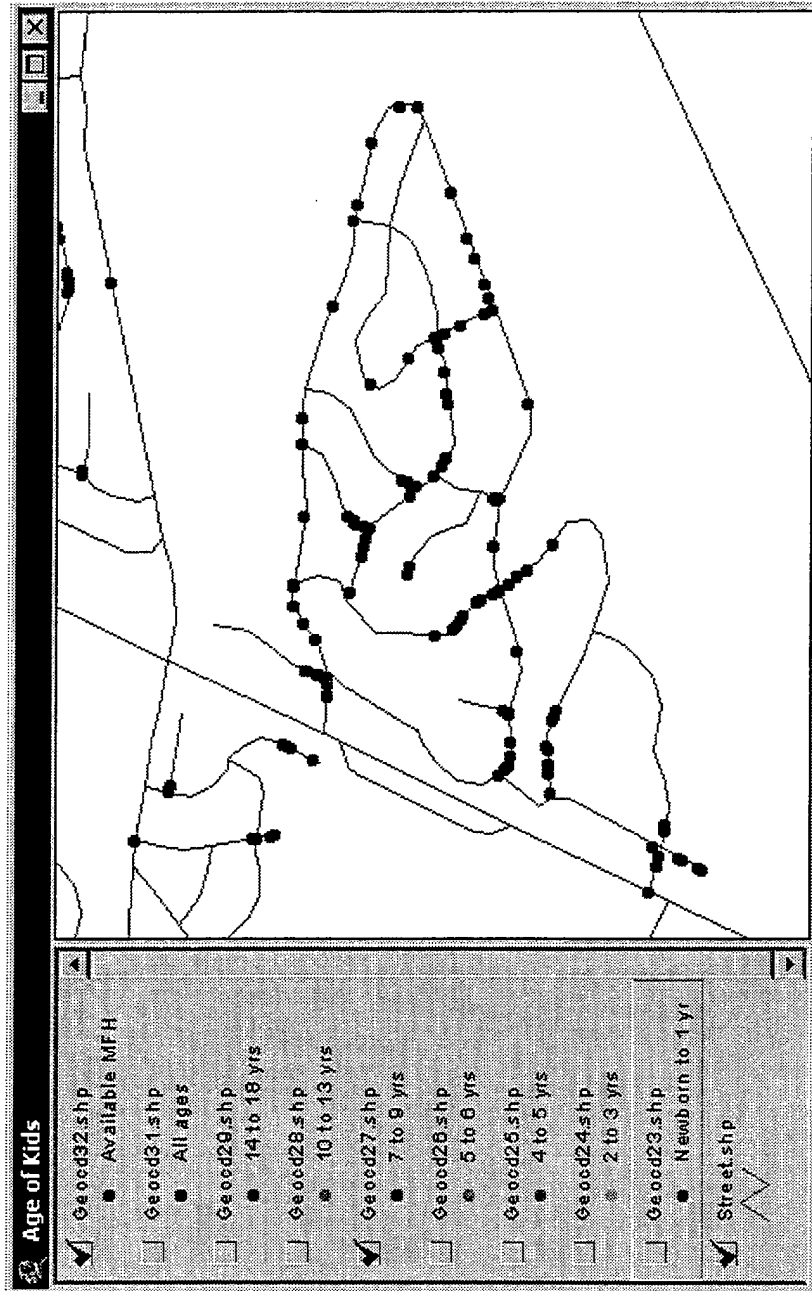
Figure 7-14.   Available Houses Superimposed with Locations of 7-9 Year Old Children

When a customer finds an available house icon surrounded by children matching his/her preferences, then s/he may want to know more specific information about the house. ArcView provides a tool, known as the identify tool, that displays table data about selected icons. For example, when a service member views the results in Figure 7-14, s/he can use the identify tool to select a particular house icon. ArcView displays the table data for the selected house, which according to the Identify Results window in Figure 7-15, is located at 240 Ardennes Circle. The window also displays other information about the house, such as the category and the floor plan code.
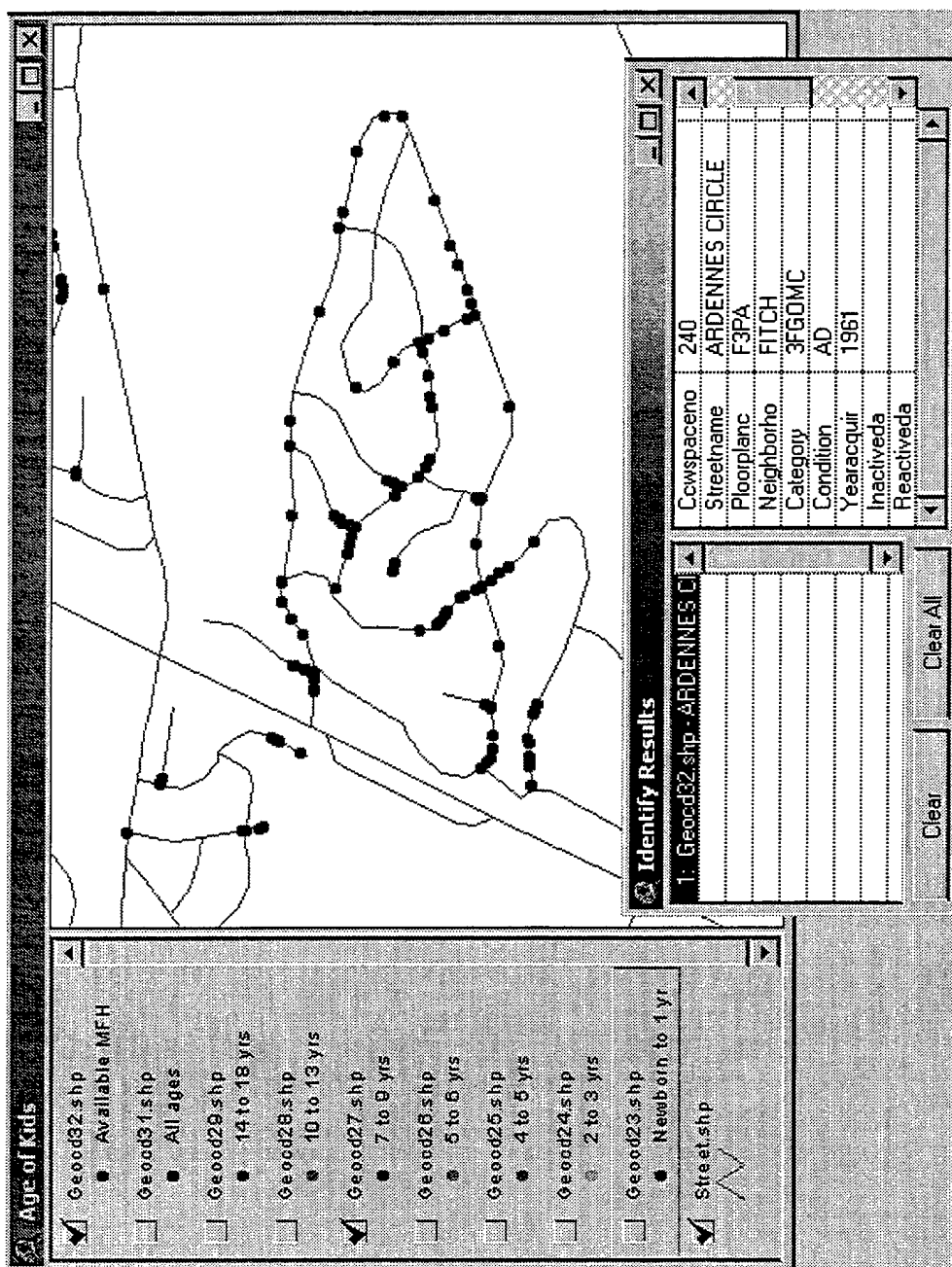
Figure 7-15.   Theme Display with Identity Results Using the Identify Tool

D. SUMMARY

Users of our HATS prototype system need to make important decisions that affect the quality of life for service members and their families. Just having tables of data or pages of reports does not necessarily improve the decision-making process. In fact, too much information, especially poorly organized information, can degrade the quality of decisions. We included decision support tools in our prototype that provide visual information that contribute to knowledge and better decisions.

Since we believe in the adage, "a picture is worth a thousand words," we chose DSTs that provide pictures for the user. Within the DBMS itself, users can view up-to-date charts depicting occupancy rates, and make decisions about the housing inventory. By implementing a GIS, we were able to capture the spatial aspect of the system. Our GIS provides dynamic pictures of houses, neighborhoods, and residents—information that would otherwise be buried in tables of data.

VIII. CONCLUSIONS AND RECOMMENDATIONS

The main purpose of this thesis was to design and build a relational database with Web connectivity and decision support tools to improve military housing utilization decisions. The ultimate goal of the system is to improve the quality of life for DoD service members and their families.

A. SUMMARY

The current housing system used at NSAMB, consisting of a legacy DBMS and a series of Microsoft Access non-relational tables, is incomplete, unreliable, and frustrating to use. We employed a top-down systems engineering design approach, starting with a detailed requirements analysis, to determine requirements for our proposed system, the Housing Assignments and Terminations System (HATS). We used the Knowledge Analysis of Tasks (KAT) method to describe the problem domain, perform a functional decomposition, and capture use cases to describe the interactions between the users and the system. Throughout the thesis, we referred to these critical requirements documents, tracing all functionality to the use cases and functional decomposition.

We then used the semantic object model (SOM) as a representation medium for creating the logical database design. The SOM for our system facilitated our understanding of the attributes of the objects and the relationships between houses, service members, assignments, and other objects in our system. The physical database design that emerged from the transformation process was normalized to avoid database anomalies.

Migrating the operational data from the current system to our prototype was the most demanding task, requiring not only a thorough understanding of the business

91

processes and the existing data schema, but also scrupulous attention to detail. A verification process must be established to ensure that the migration provides valid data in the new system. Further, data managers should educate personnel concerning the best ways to capture and record data. In addition, they should encourage personnel to implement data definition standards whenever possible, making future migrations more reasonable.

Since computers are ubiquitous in today's business offices, users expect effective and pleasing GUIs with their business systems. While there are a variety of powerful applications on the market, acquisition and training costs are additional factors to consider for system implementation. We implemented our prototype using applications familiar to the users and already owned by the Housing Welcome Center (HWC). We created forms, reports, and other views for our prototype using Microsoft Access. In addition, we created and published an online housing application form, DD 1746, using Microsoft FrontPage and Internet Information Server (IIS). By implementing ASP and ODBC technology in our system, customers have direct input to the database, eliminating the need for the HWC to re-type housing applications. For our prototype, Microsoft's family of applications provides a pleasing and functional GUI, with acceptable system performance.

Data obscurity degrades the ability of humans to make decisions. Our prototype includes visual decision support tools (DSTs) that improve the quality of housing utilization decisions made by the HWC and its customers. Access provides charts and other visual tools that help organize data into useful information. A geographic information system (GIS), such as ESRI's ArcView, captures and displays the spatial

aspect of HATS. By implementing DSTs in our prototype, we provide users with visual information that contribute to knowledge and better decisions.

B. RECOMMENDATIONS

Based on the results of our research, we make the following recommendations:

- Before implementing any features of the prototype, the HWC should verify the quality of the operational data in the prototype. Although we performed an extensive data migration process, we urge housing personnel more familiar with the specific details of the assignment and termination process to verify data in the prototype. In addition, since we took a snapshot of the current database to perform the migration, we recommend updating the prototype's database to reflect changes in the operational data. For example, we froze the current database on 18 April 2001. Since then, the HWC has continued to enter and modify data in the current system; however, these changes have not been made to the prototype.

- Implement the online housing application form with connectivity to the database. Modify the prototype Web site as needed and use the Access forms to allow customers to submit applications directly into the HWC database. The current system requires HWC counselors to re-type applications sent by service members via the Web application. The prototype allows the counselors to simply review customer-entered applications and click a button to accept them, eliminating redundant data entry.

- Replace the current non-relational Access database with a relational database—such as the prototype database. The HWC Access database is not designed to use the full potential of a DBMS. It allows data duplication, restricts data integration, and fails to enforce business rules and constraints. The prototype's relational database is designed to overcome these limitations and improve data quality.

- Review the policies and requirements to maintain an off-post rental referral service. Ten years ago it made sense to track off-post rental property and maintain an information system to store and report rental information to service members. However, in the Internet-age, service members can view the same information, if not better information, via non-HWC Web sites. Consider sharing information and forming partnerships with off-post referral agencies to reduce the cost and effort to provide a rental referral service. At a minimum, the HWC can provide Internet-enabled terminals and recommended Web addresses to assist service members with off-post housing needs.

- Improve the prototype and consider full implementation to take advantage of the decision support tools (DSTs). Seize the opportunity to improve the prototype by working with students and faculty at the Naval Postgraduate

School, as well as analysts at TRAC-Monterey. A fully developed GIS can provide stimulating visual information to improve decisions made by the HWC and its customers.

C.   RECOMMENDATIONS FOR FURTHER DEVELOPMENT

The following are identified as opportunities for further development:

- Enhance the prototype by improving the GUI and GIS. Consider improving the GUI by using more robust programming languages, such as Java or C+. Design a networked GIS and link the system to additional sources of data layers, such as local law enforcement agencies and environmental organizations.

- Implement the prototype in a networked environment. Determine the most appropriate applications for implementing a "full up" network system. Consider fine-tuning the prototype database in Access for increased performance. This could include making the database file an executable file (with an .mde extension), or separating the forms, reports, and macros from the data files. Also consider implementing the database using other applications, such as Oracle. Compare the alternatives based on functionality, network performance, training required, and cost.

- Determine if commercial off-the-shelf (COTS) software is a viable alternative to implement the Assignment and Termination System. Consider the advantages and disadvantages of using COTS software.

- Integrate the prototype with other NSAMB data systems. Consider data sharing initiatives to provide a logically unified database within NSAMB. By using shared common data definitions, activities within NSAMB can share service member information, reducing inconsistencies and redundancies. For example, design an architecture to connect applications used by the installation vehicle registration, childcare services, recreation activity, and the registrar.

94

APPENDIX A. USE CASES

Use Case Name	UC 5. Apply Online
Participating Actor	Housing Web Server (HWS) SM (through an Internet Browser)
Summary	The SM visits the NSAMB Housing Web site. She fills out the DD Form 1746, Application For Assignment To Housing, by entering personal information in the required fields of the Web page.
Basic Course Of Events	1. A military service member selects the "Apply For Housing Online" function on the Housing Web page. 2. The HWS responds by presenting a blank DD Form 1746, Application For Assignment To Housing, to the SM. 3. The SM completes the required fields: type of service (military housing or referral), name, SSN, pay grade, DOD component, address (street, city, zip), home phone, duty phone, status of applicant, marital status, separated from dependents, request housing for, origin organization, destination organization, effective rank date, active duty service computation date, date remaining on active duty, effective date change in duty location, report date, estimated family arrival date, dependents' names, dates of birth, sex, relationship. If the SM is only asking for referral information, then she selects the type of housing desired, amenities desired, number of bedrooms, number of bathrooms, date housing needed, price range, location preference, and remarks. After completing all required fields, she signifies completion of the action. 4. The HWS responds by activating the CGI script that emulates the "Add Service Member Record" function in the system. The HWS adds the record to the HATS database. 5. HATS sends an acknowledgement to the HWS. 6. The HWS sends and acknowledgement to the SM (on the Internet).
Alternative Paths	None.

Exception Paths	In step 2, if the SM enters invalid data, or if the user leaves required fields blank, the Web server responds by notifying the SM that the input was incorrect and asking her to correct the DD Form 1746.
Triggers	The SM has orders to the area serviced by the Housing Office and desires a future housing assignment.
Assumptions	The Web server will not accept a DD Form 1746 if the SM does not enter valid information. The system does not need to inform the SM of incorrect field entries.
Preconditions	The Web server is operational.
Postconditions	The SM is added to the expected waiting list.
Related Business Rules	NSAMBINST 11101.4: eligibility requirements, housing categories, last assignment, and dependents.
Special Requirements	The HWS receives the acknowledgement within 30 seconds. Depending on the Service Member's Internet connection, she receives the acknowledgement not later than 60 seconds after clicking the "Submit" button on her browser.

Use Case Name	**UC 10. Apply For Housing**
Participating Actor	Housing Management Specialist (HMS)
Summary	The SM applies for housing in person, or through the mail. She fills out the DD Form 1746, Application For Assignment To Housing, and gives a hard copy to the HMS.
Basic Course Of Events	1. A SM submits a completed DD Form 1746 to the HMS, who in turn activates the "Add SM Record" function of her terminal. 2. HATS responds by presenting the DD Form 1746 on the display to the HMS. 3. The HMS completes the fields, by selecting or entering the following: type of service (military housing or referral), name, SSN, pay grade, DOD component, address (street, city, zip), home phone, duty phone, status of applicant, marital status, separated from dependents, request housing for, origin organization, destination organization, effective rank date, active duty service computation date, date remaining on active duty, effective date change in duty location, report date, estimated family arrival date, dependents' names, dates of birth, sex, relationship. If the SM is only asking for referral information, then she selects the type of housing desired, amenities desired, number of bedrooms, number of bathrooms, date housing needed, price range, location preference, and remarks. After completing all required fields, she signifies completion of the form. 4. HATS acknowledges the addition of a SM to the database and adds the SM to the expected waiting list. 5. The HMS receives the acknowledgement and has the option of viewing the SM on the waiting list.
Alternative Paths	In step 4, the SM gives the HMS verification of dependents, orders, and departure from last assignment. This alternative path puts the SM on the actual waiting list.
Exception Paths	None.

Triggers	The SM has orders to the area serviced by the Housing Office and desires a housing assignment.
Assumptions	None.
Preconditions	The SM wants government housing.
Postconditions	The SM is added to the waiting list.
Related Business Rules	NSAMBINST 11101.4: eligibility requirements, housing categories, last assignment, and dependents.
Special Requirements	The SM is not added to the database until the HMS clicks on the "Submit" button. Once added, HATS sends an acknowledgement within 15 seconds.

Use Case Name	**UC 15. Modify SM**
Participating Actor	HMS
Summary	The HMS may need to correct inaccurate or outdated SM information such as number of dependents or an incorrect SSN.
Basic Course Of Events	1. The use case begins when the HMS selects the "Edit SM Record" function of HATS. 2. HATS responds by prompting the HMS to select the SM to modify. 3. The HMS selects the SM. 4. HATS displays the SM's record. 5. The HMS modifies the record. 6. HATS displays the updated information.
Alternative Paths	In step 2, the HMS can select the SM by entering name, SSN, or address (if already assigned).
Exception Paths	In step 5, the HMS enters duplicate information, such as SSN. HATS alerts the HMS and prompts her to enter valid information. The HMS enters the correct information.
Triggers	Someone notifies the HMS that the SM information is not correct, prompting the HMS to modify the SM's record.
Assumptions	None.
Preconditions	The SM is in the system.
Postconditions	The SM's record is updated.
Related Business Rules	None.
Special Requirements	None.

Use Case Name	UC 20. View and Print Waiting Lists
Participating Actor	HMS
Summary	The Housing Office constantly views the waiting list to update SMs on their place on the list and their expected availability date. The HMS's also print periodic lists to display in the Housing lobby.
Basic Course Of Events	1. The HMS activates the "View/Print Waiting Lists" function of her terminal. 2. HATS responds displaying all of the different waiting lists and prompts the HMS to select one or more list(s) to view. 3. The HMS selects the lists. 4. HATS displays the lists on the HMS terminal and prompts the HMS to print the list(s). 5. The HMS selects the option to print. 6. The HMS receives the desired reports and makes her selection to navigate to other menus.
Alternative Paths	In step 3, the HMS selects to print a list without viewing it first. The system skips steps 4 and 5.
Exception Paths	None.
Triggers	A SM calls or visits the HMS to ask for an expected move-in date. Every morning, an HMS prints the lists to display in the lobby.
Assumptions	None.
Preconditions	The list is already sorted according to the business rules.
Postconditions	None.
Related Business Rules	NSAMBINST 11101.4: freeze zones and waiting lists.
Special Requirements	All sub-functions end with a menu to navigate to the main "Manage Waiting Lists" function, or back to the HATS main menu.

Use Case Name	UC 25. Move SM to Actual Waiting List
Participating Actor	HMS
Summary	The SM gives the HMS verification of dependents, orders, and departure from last assignment. With this information, the HMS is authorized to move the SM to the actual waiting list.
Basic Course Of Events	1. The HMS selects the "Add SM to Actual Waiting List" function in HATS. 2. HATS responds by prompting the HMS to select the SM, by SSN or by last name. 3. The HMS selects the SM. 4. HATS displays the SM and the waiting list category (that the SM is currently on in the expected waiting list) and prompts the user to "Move SM to the Actual Waiting List". 5. The HMS signifies that she wants to move the SM to the actual list. 6. HATS sends an acknowledgement to the HMS.
Alternative Paths	In step 1, the HMS selects the SM from the expected waiting list and signifies that she wants to move the SM to the actual list, skipping steps 2-5.
Exception Paths	None.
Triggers	The SM desires government housing and presents the HMS with valid prerequisite information.
Assumptions	None.
Preconditions	The SM is already on the expected waiting list.
Postconditions	The system removes the SM from the expected list and places her on the actual waiting list (and assigns the SM a control date).
Related Business Rules	NSAMBINST 11101.4: departure date eligibility, DEERS, and orders.
Special Requirements	The HMS has the option of viewing the new waiting list for the SM. HATS sends an acknowledgement within 30 seconds.

Use Case Name	**UC 30. Promote SM**
Participating Actor	HMS
Summary	The SM receives a promotion since she entered the expected waiting list, making her eligible for a higher category of housing. The HMS moves the SM to the new actual waiting list.
Basic Course Of Events	1. The HMS selects the "Add SM to Actual Waiting List" function in HATS. 2. HATS responds by prompting the HMS to select the SM, by SSN or by last name. 3. The HMS selects the SM. 4. HATS displays the SM and the waiting list category (that the SM is currently on in the expected waiting list) and prompts the user to "Move SM to the Actual Waiting List". 5. The HMS signifies that she wants to move the SM to the actual list. 6. HATS sends an acknowledgement to the HMS.
Alternative Paths	None.
Exception Paths	None.
Triggers	The SM presents proof of promotion to the HMS and requests a move to the new category waiting list.
Assumptions	None.
Preconditions	The SM is already on an actual waiting list.
Postconditions	The SM is on a different actual waiting list that matches her rank or grade. The SM is no longer on the previous waiting list.
Related Business Rules	NSAMBINST 11101.4: promotion status of applicant.
Special Requirements	The HMS has the option of viewing the new waiting list for the SM. HATS sends an acknowledgement within 30 seconds.

Use Case Name	UC 35. Delete SM From Waiting List
Participating Actor	HMS
Summary	A SM is on either the expected or actual waiting list, but has decided to live off base.
Basic Course Of Events	1. The HMS selects the "Delete SM from Waiting List" function in HATS. 2. HATS responds by prompting the HMS to select the SM, by SSN or by last name. 3. The HMS selects the SM. 4. HATS displays the SM and prompts the user to "Delete SM from Waiting List". 5. The HMS signifies that she wants to delete the SM from the list. 6. HATS sends an acknowledgement to the HMS.
Alternative Paths	In step 1, the HMS selects the SM from either the expected or actual waiting list and signifies that she wants to remove the SM from the list, skipping steps 2-5.
Exception Paths	None.
Triggers	A SM requests that the HMS remove her from the waiting list.
Assumptions	None.
Preconditions	The SM is on a waiting list.
Postconditions	The SM's personal information stays in the system for future assignments.
Related Business Rules	NSAMBINST 11101.4: removing SM's from waiting lists.
Special Requirements	The HMS should be able to navigate directly from this function to the off base referral function. HATS sends an acknowledgement within 30 seconds.

Use Case Name	UC 40. Maintain Freeze Zone
Participating Actor	HMS
Summary	Housing stabilizes the position of the top ten percent of SMs on each actual waiting list by not placing new arrivals in the freeze zone, regardless of rank, duty assignment, or detachment date. They make exceptions for key and essential SMs, based on specific guidance. When this happens, Housing places these SMs at the top of the freeze zone, without displacing SMs with a housing offer.
Basic Course Of Events	1. The HMS activates the "Maintain Freeze Zone" function. 2. HATS displays all of the different categories of waiting lists and prompts the HMS to select one or more list(s) to maintain. 3. The HMS selects the "Regenerate Sequence Numbers" function. 4. HATS performs the function and displays the waiting list(s) indicating the freeze zone. 5. HATS sends an acknowledgement to the HMS.

Alternative Paths	In step 3, the HMS selects the "Change Freeze Zone Percentage" function. HATS displays the current percentage and prompts the HMS to change it. The HMS enters the new percentage.
	    HATS regenerates the Freeze Zone based on the new percentage and sends an acknowledgement to the HMS.
	    In step 3, the HMS places a key SM at the top of the freeze zone.
	    In step 4, the HMS has the option to print the list(s).

Exception Paths	None.
Triggers	A key SM arrives and requests housing. Housing decides to decrease turbulence and increase the freeze zone. Housing decides to regenerate sequence numbers and update the freeze zone (usually after several SMs have moved off the freeze zone and into housing).

Assumptions	None
Preconditions	None
Postconditions	If a SM moves into housing, the system removes her from the freeze zone (actual waiting list), but does not automatically move new SMs to fill her position. If the HMS regenerates the sequence numbers, she creates a new zone based on the freeze zone percentage.
Related Business Rules	NSAMBINST 11101.4: freeze zone.
Special Requirements	HATS sends the acknowledgement to the HMS in no more than 30 seconds.

Use Case Name	**UC 45. Offer Housing**
Participating Actor	HMS
Summary	The HMS notifies a SM that her sequence number is next on the actual waiting list. The system finds three houses matching the SM's category and the HMS offers the SM to choose one house to live in (after visual inspection).
Basic Course Of Events	1. The HMS selects the "Make Housing Offer" function. 2. HATS responds by prompting the HMS to select the SM. 3. The HMS selects the SM by name or SSN. 4. HATS displays the SM's name, SSN, command, and eligibility category for verification. HATS prompts the HMS to verify the information and begin the search. 5. The HMS signifies that she wants to begin the search. 6. HATS responds by listing three housing addresses that match the search. HATS prompts the HMS to print a copy of the offer. 7. The HMS signifies that she wants to print the offer and she makes a formal offer to the SM.
Alternative Paths	In step 2, the HMS can also enter a specific address if it is a direct assignment. The HMS signifies that she wants to directly assign a SM, invoking the Make Assignment use case.
Exception Paths	In step 4, the HMS notices incorrect information relating to the SM. She makes the corrections and continues with the offer.
Triggers	The SM moves to the top of the actual waiting list and the HMS meets with the SM to make an offer.
Assumptions	The HMS only offers available houses, as reported by the HFS.

Preconditions	The SM has the next sequence number on the actual waiting list.
Postconditions	The system list three houses to offer the SM.
Related Business Rules	NSAMBINST 11101.4: offering housing--If housing cannot contact the SM with the next sequence number, the HMS may offer MFH to the next SM on the list. The HMS can skip the SM as long as she cannot be reached, and eventually, the SM can be placed at the bottom of the waiting list.
Special Requirements	HATS responds to the HMS within 30 seconds of the search request.

Use Case Name	UC 50. Make Assignment
Participating Actor	HMS
Summary	The SM chooses to occupy one of the houses offered by the HMS, who makes the official assignment (matching the SM with a housing address).
Basic Course Of Events	1. A SM accepts an offer (after selection or direct assignment) and the HMS selects the "Assign Housing" function in HATS. 2. HATS prompts the HMS to select the SM (by name or SSN). 3. The HMS enters name or SSN. 4. HATS displays the SM's name, rank, SSN and the addresses offered to the SM. HATS prompts the HMS to verify the information and proceed. 5. The HMS signifies that she wants to proceed with the assignment. 6. HATS creates the assignment and prints a Housing Acceptance Letter.
Alternative Paths	In step 2, the HMS searches for the address chosen by the SM to occupy. HATS displays the address and the name, rank, and SSN of the SM offered the house.
Exception Paths	In step 2, the HMS enters the wrong address and the system responds with an existing assignment or with no offer. HATS displays an error message and the HMS enters another address.
Triggers	The SM tells the HMS that she accepts an offer for housing, causing the HMS specify an assignment in the system.
Assumptions	None
Preconditions	The SM accepted a housing offer.
Postconditions	HATS makes an assignment relating the SM and the chosen housing address.
Related Business Rules	NSAMBINST 11101.4: assignment and inspection.
Special Requirements	The HMS can only enter addresses already in the system.

Use Case Name	UC 55. Terminate Assignment
Participating Actor	Initiated by HMS Communicates with Facilities Supervisor (HFS)
Summary	SMs frequently receive orders to move to other duty stations. Once a SM has orders, she completes an "Intent to Vacate" letter and informs the HMS of the specific dates. Eventually, the SM moves and terminates the housing assignment.
Basic Course Of Events	1. The HMS selects the "Terminate Housing" function of HATS. 2. HATS responds by prompting the HMS for the SM name or SSN. 3. The HMS chooses the departing SM. 4. HATS displays the SM's name, SSN, and unit address for verification and prompts the HMS to enter the expected vacancy date. . 5. The HMS enters the date. 6. HATS displays an "Intent to Vacate" letter, displaying the SM's name and SSN, MFH address, a tentative pre-termination inspection date, and the expected vacancy date (which will become the final inspection date). HATS prompts the HMS to verify the information, change dates if necessary, and print the "Intent to Vacate" letter. 7. The HMS verifies (and changes dates if necessary) and signifies that she wants to print the letter. 8. After the final inspection (on the expected vacancy date), the SM returns to the HMS with the clearing sheet (and the inspector's signature). The HMS navigates the "Terminate Housing" menu to select the "Print Termination Letter" function. 9. HATS prompts the HMS to enter the SM's name or SSN. 10. The HMS enters the name. 11. HATS displays the SM and address information. 12. The HMS signifies that she wants to print the termination letter and she terminates the housing assignment. 13. HATS prints a termination letter for the SM

	and changes the status of the MFH to unavailable.
Alternative Paths	In step 2, the HMS enters the soon-to-be-terminated housing address. In steps 9 and 10, the HMS enters the address.
Exception Paths	In step 4, the HMS enters a past date, indicating that the SM has already moved. HATS responds by sending an error message and prompting the HMS to enter another date.
Triggers	A SM has orders and requests that the HMS terminate her housing assignment.
Assumptions	The expected vacancy date is less than one year from the current date.
Preconditions	The SM is assigned housing.
Postconditions	The SM is not assigned housing and is eligible for Basic Allowance for Housing.
Related Business Rules	NSAMBINST 11101.4: terminations, inspections, and outprocessing.
Special Requirements	None.

Use Case Name	**UC 60. Prepare CRL**
Participating Actor	HMS
Summary	An incoming SM wants information on off-base housing. The SM gives the HMS search criteria for desired living arrangements. The HMS enters the information into HATS and prints a Customer Referral List for the SM.
Basic Course Of Events	1. The HMS selects the "Prepare Customer Referral List" function of HATS. 2. HATS responds by prompting the HMS to enter the SM's name or SSN. 3. The HMS enters the name or SSN. 4. HATS prompts the HMS to select search criteria (price, size, location, smoking, pets, number of matches to display, and so on). 5. The HMS indicates the SM's preferences. 6. HATS displays the results and prompts the HMS to print the CRL. 7. The HMS signifies that she wishes to print the CRL. 8. HATS prints the CRL.
Alternative Paths	None.
Exception Paths	In step 6, the HMS enters the SM's preferences and HATS does not find any matching referrals. HATS sends the HMS an error message and prompts her to enter less restrictive preferences.
Triggers	A SM asks the HMS to find off-base housing matching her preferences.
Assumptions	The SM can already be assigned MFH. This is a one-time transaction and the HMS does not keep a record of the details of each referral. The SM asks for a one-time referral. She can always ask for a new one in the future, but the HMS has to enter new search criteria—the system does not save CRLs or search criteria.

Preconditions	HATS already knows the SM and the private property.
Postconditions	None.
Related Business Rules	None.
Special Requirements	None.

Use Case Name	**UC 65. Add or Edit MFH**
Participating Actor	HMS
Summary	Occasionally, the HMS needs to edit the description of MFH to reflect upgrades or modifications. In the event of new construction, this is the use case used to add the address to the inventory.
Basic Course Of Events	1. This use case begins when the HMS selects the "Add or Modify Unit Details" function of HATS 2. HATS responds by prompting the HMS for the address of the unit to modify. 3. The HMS enters the address. 4. HATS displays a description of the MFH. 5. The HMS changes any desired field or fields and signifies that she is finished with the editing. 6. HATS accepts the changes and displays the address with the most current information.
Alternative Paths	In step 3, the HMS signifies that she wishes to add a new address. HATS displays blank fields for the HMS to enter information describing the quarters. The HMS signifies that she is finished and wishes to exit the screen. HATS acknowledges the addition of a new address to the system.
Exception Paths	None.
Triggers	The HMS wishes to change information about an address or add a new address to the system.
Assumptions	None.
Preconditions	If the HMS wishes to edit an address, the address must already exist. If the HMS wishes to add a new address, the address cannot already exist in HATS.
Postconditions	None.
Related Business Rules	None.
Special Requirements	None.

Use Case Name	UC 70. Add or Edit Private Property
Participating Actor	HMS
Summary	Details about off base rental property change almost daily. The HMS needs to add new rental property to the system and change descriptions and terms to existing properties.
Basic Course Of Events	1. The use case begins when the HMS selects the "Manage Referral Information" function of HATS. 2. HATS prompts the HMS to select the property to edit. 3. The HMS selects the property. 4. HATS displays the property information. 5. The HMS edits all desired fields and signifies that she is finished. 6. HATS acknowledges the editing and displays an updated view of the property.
Alternative Paths	In step 2, the HMS selects the owner or property manager. HATS displays the addresses of the properties owned or managed and prompts the HMS to select one. The HMS selects an address. In step 2, the HMS signifies that she wishes to add a new rental property to the system. HATS responds by asking the HMS to enter information describing the property. The HMS enters the information. HATS acknowledges the addition of a new property.
Exception Paths	In step5, the HMS enters a new property owner or manager who does not exist in the system. HATS alerts the HMS that she must also enter information about the owner or manager.
Triggers	The HMS has to change information about rental property due to new rental terms or characteristics of the property. An owner or manager lists new property with the HMS.
Assumptions	None.

114

Preconditions	If the HMS wishes to edit property information, the address must already exist in the system. If the HMS wishes to add the property to the system, it cannot already exist.
Postconditions	None.
Related Business Rules	NAVINST 1101.04: Off base referrals and rules governing rental property.
Special Requirements	None.

Use Case Name	UC 75. Delete Address
Participating Actor	HMS
Summary	For one reason or another, the HMS needs to remove either government or privately-owned property. For example, the owner of an off base property decides to sell and no longer offer it for rent. Or in the case of the government quarters, the base sells the property to another entity.
Basic Course Of Events	1. The use case begins when the HMS selects the "Delete an Address" function of HATS. 2. HATS prompts the HMS to enter the address. 3. The HMS enters the address. 4. HATS displays the address details and prompts the user to verify the deletion. 5. The HMS verifies the deletion. 6. HATS deletes the record of the address and returns to the "Manage Housing Inventory" menu.
Alternative Paths	In step 2, the HMS signifies that she wishes to delete an off base address. In step 6, HATS displays the "Manage Referral Information" menu.
Exception Paths	None.
Triggers	The HMS needs to delete the record of either a government or a privately-owned property.
Assumptions	None.
Preconditions	The address must already exist.
Postconditions	HATS deletes the record of the address. Records of historical assignments are not deleted.
Related Business Rules	None.
Special Requirements	None.

116

Use Case Name	**UC 80. Add or Edit Owner or Manager**
Participating Actor	HMS
Summary	The HMS adds or edits information about an owner or property manager. This can happen due to a sale, or when a new manager takes over the property.
Basic Course Of Events	1. The use case begins when the HMS selects the "View Referral Details" function of HATS. 2. HATS prompts the HMS to enter the owner or manager's information. 3. The HMS enters the information, signifying whether she's an owner or manager, and selecting the related property (if known at the time). 4. HATS acknowledges the addition.
Alternative Paths	In step 2, the HMS signifies that she wishes to edit owner or manager information. HATS prompts her to select the person to edit. The HMS edits the information and signifies that she is finished. HATS acknowledges the editing.
Exception Paths	None.
Triggers	An owner or manager informs the HMS that there has been a change in ownership or management.
Assumptions	None.
Preconditions	In order to edit the owner or manager, she must already exist in the system.
Postconditions	The owner or manager is in the system. If the HMS just edits the owner or manager, then the new information is saved.
Related Business Rules	None.
Special Requirements	None.

Use Case Name	UC 85. Change Availability
Participating Actor	Facilities Supervisor (HFS)
Summary	When a specific house is unavailable to SMs, the HFS has to provide a date when she thinks the unit will be ready for occupancy. Once move-out or other maintenance is complete, the HFS notifies the system that the MFH is fit for a new occupant.
Basic Course Of Events	1. The HFS selects the "Manage Availability of MFH" function within HATS. 2. HATS prompts the HFS to enter the MFH address. 3. The HFS enters the address. 4. HATS displays the address information on the HFS terminal and prompts the HFS to enter or change the expected availability date, or make the address available. 5. The HFS enters the expected date, or signifies that she wants to make the address available, and signifies that she wants to proceed. 6. HATS displays the current availability information.
Alternative Paths	None.
Exception Paths	In step 5, the HFS enters a future date for the "actual available date" field. HATS alerts the HFS that it only accepts past or current dates in that field. HATS returns to the field and prompts the HFS to enter another date.
Triggers	Termination of MFH, maintenance factors affect the expected availability date, or the unit becomes available.
Assumptions	The HFS enters expected and actual availability dates within a few days of the discovery.
Preconditions	The MFH is in the system and unoccupied.
Postconditions	HATS knows the expected available date or the unit is available.
Related Business Rules	None.

Special Requirements	The HFS does this function remotely. This may require a CGI script or other interface to change the availability of housing. (HATS updates the availability status in the database.)

Use Case Name	**UC 90. View and Print Reports**
Participating Actor	HMS HFS
Summary	There are over 20 reports that users request from the system on any given day. The HMS or HFS can use the "Queries & Reports" function of HATS or she can use menus in the other functions to view and print reports.
Basic Course Of Events	1. The use case begins when the actor selects the "Queries & Reports" function of HATS. 2. HATS responds by prompting the actor to select a specific report to view or print. 3. The actor selects the desired report. 4. HATS displays the report and prompts the actor to print the report. 5. The actor signifies that she wants to print the report. 6. HATS prints the report.
Alternative Paths	In step 1, the actor can also select the view or print report function from within the other functions, such as "Assignments & Terminations". In step 3, the actor can print the report without viewing it. In step 6, HATS returns to the report view and prompts the actor to exit the report and return to the "Reports" main menu. The actor signifies that she wants to exit the menu. HATS returns to the "Reports" main menu.
Exception Paths	None.
Triggers	The actor needs to print a daily, weekly, monthly, quarterly, annual, or other report.
Assumptions	None.
Preconditions	The report must already exist.
Postconditions	None.
Related Business Rules	NSAMBINST 11101.4: reporting procedures and waiting list reports.

Special Requirements	Internet users can also view reports by using the Web menu to select the desired reports. The interaction is the same as this use case, except there are fewer reports available online. In either case, HATS displays the selected reports within 30 seconds of the user's request.

Use Case Name	UC 95. Manage Accounts
Participating Actor	HDBA (Housing Database Administrator)
Summary	Employees come and go, get promoted, or change positions at the Housing Office, requiring changes to user access to the system.
Basic Course Of Events	1. The use case begins when the HDBA selects the "Manage User Access" function of HATS. 2. HATS responds by prompting the HDBA to select a user. 3. The HDBA selects the user. 4. HATS displays the user's access permissions. 5. The HDBA changes the permissions as desired. 6. HATS displays the new permissions.
Alternative Paths	In step 2, HATS asks if the HDBA wants to "Add a New User". The HDBA replies that she wants to add a new user. HATS displays an input form for the HDBA to enter the user's information. In step 5, the HDBA signifies that she wants to delete the user from the system. HATS prompts her for a confirmation. The HDBA confirms the deletion. HATS acknowledges the action.
Exception Paths	In step 2, the HDBA adds a new user, but she enters a duplicate SSN. HATS alerts the HDBA and prompts her to enter valid data. The HDBA enters a valid SSN.
Triggers	A new employee needs access to the system or a former employee no longer needs access to the system.
Assumptions	None.
Preconditions	If the HDBA wants to modify or delete a user, she must already exist in the system. If the HDBA wants to add a user, she cannot already be in the system.
Postconditions	If the HDBA adds a new user, the system creates a new user record. If the HDBA deletes a user, the user no longer has access. If the HDBA modifies user information, such as access, the user has the updated permissions.
Related Business Rules	None.

Use Case Name	**UC 100. Maintain Database**
Participating Actor	HDBA
Summary	The HDBA needs to maintain the database in the system occasionally. She may need to change forms, reports, or other internal files in order to maintain the system. Also, Housing does not want another user inadvertently changing the database structure and corrupting the system.
Basic Course Of Events	1. The use case begins when the HDBA selects the "Manage Database" function of HATS. 2. HATS responds by prompting the HDBA to select the forms, tables, or reports to modify. 3. The HDBA selects the item to modify. 4. HATS displays the item. 5. The HDBA performs maintenance and closes the screen.
Alternative Paths	None.
Exception Paths	None.
Triggers	The HDBA needs to change a form or report in the system.
Assumptions	None.
Preconditions	None.
Postconditions	None.
Related Business Rules	None.
Special Requirements	This is not an essential use case. Because we already know that we're using MS Access, we don't need to design a fictional interface to the data.

Use Case Name	UC 105. Authenticate User
Participating Actor	HMS HFS
Summary	All users who request entry to the system must prove they are who they say they are (authentication). This applies to users inside Housing as well as outside. Who the users are determines what they have access to (authorization).
Basic Course Of Events	1. The use case begins when the user enters a user identification and password. 2. The system responds by checking the combination of user identification and password against the recorded list of valid users. 3. The system responds by notifying the user that the user identification and password are valid and allows the user access to the system.
Alternative Paths	None
Exception Paths	In step 2, if the user identification and password combination is not valid, the system responds by notifying the user that the combination was invalid and asks the user to try again. If the user tries three times unsuccessfully, the system responds by disallowing the user and marking as suspended the user record of any and all user identifications tried. In step 2, if the user identification that is tried is in suspended mode, the system responds by notifying the user that this identification has been suspended and the user must contact the system administrator.
Triggers	The user requires access to the system.
Assumptions	If the system resides on a network system, the user already has access to the network.
Preconditions	The system is operational.
Postconditions	The validated user is allowed access to the system.
Related Business Rules	None.

APPENDIX B.   HATS SEMANTIC OBJECT MODEL

SERVICEMEMBER Object

- Key is SSN;
- SM may currently occupy at most one MFHUNIT (currently assigned to);
- SM may be enrolled in only one WAITING LIST;
- SM may not have FAMILY MEMBERS (in the case of a referral), but he may have many;
- SM is not required to have a current housing assignment, but he may have at most one.

MFHUNIT Object

- Key is UnitID;
- MFH may have at most one SM;
- MFH is on one-and-only-one BASE;
- MFH is in one-and-only-one NEIGHBORHOOD;
- MFH has one-and-only-one FLOOR PLAN;
- MFH is described by one-and-only-one HOUSING CATEGORY;
- MFH has 0-to-many APPLIANCES;
- MFH can have only one ASSIGNMENT at any point;

WAITINGLIST Object

- Key is WaitingListTitle (BASE + HOUSING CATEGORY);
- WAITING LIST may have 0-to-many SERVICE MEMBERS waiting for housing.

ASSIGNMENT Object

- Key is AssignmentID (SERVICE MEMBER SSN + MFH UNIT Address + AssignmentDate);
- ASSIGNMENT can be made by one-and-only-one HMS.

FAMILYMEMBER Object

- Key is FamilyMemberID (SM SSN + RelationCode);

FLOORPLAN Object

- Key is FloorPlanCode;
- A FLOOR PLAN may be used in many different MFH UNITS;

HOUSINGCATEGORY Object

- Key is Designation;
- A HOUSING CATEGORY can apply to many MFH UNITS;
- A HOUSING CATEGORY can have at most one WAITING LIST.

APPLIANCE Object

- Key is ApplianceID;
- An APPLIANCE may be in only one MFH;

BASE Object

- Key is BaseName;
- A BASE may have one-to-many NEIGHBORHOODS;
- A BASE may have one-to-many MFH UNITS.

NEIGHBORHOOD Object

- Key is NeighborhoodName;
- A NEIGHBORHOOD may be on one-and-only-one BASE;
- A NEIGHBORHOOD may contain one-to-many MFH UNITS.

PRIVATE RENTAL PROPERTY Object

- Key is Address;
- A PRIVATE RENTAL PROPERTY can only have one OWNER;

OWNER Object

- Key is OwnerCode;
- An OWNER may have one-to-many PRIVATE RENTAL PROPERTIES.

SYSTEM USER Object

- Key is UserID;
- A SYSTEM USER may also be an HMS (subtype);
- A SYSTEM USER may also be an HFS (subtype);
- A SYSTEM USER may also be a SYSTEM ADMINISTRATOR (subtype).

HMS Object

- An HMS is a subtype of SYSTEM USER with the same key of UserID;
- An HMS may create many ASSIGNMENTS;
- An HMS has one-and-only-one access level.

HFS Object

- An HFS is a subtype of SYSTEM USER with the same key of UserID;
- An HFS has one-and-only-one access level.

SYSTEM ADMISTRATOR Object

- A SYSTEM ADMINISTRATOR is a subtype of SYSTEM USER with the same key of UserID;
- A SYSTEM ADMINISTRATOR has one-and-only-one access level.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C.   HATS RELATIONAL MODEL

SERVICEMEMBER (SSN, LastName, FirstName, MI, Paygrade, BranchOfService, DateOfRank, PRD, NumberOfFamilyMembers, Command, Priority, HomePhone, WorkPhone, SpouseWorkPhone, Email, BedRoomsRequested, BedRoomsAuthorized, DeferUntilDate, ControlDate, Pets, Weapons, Medical, Remarks, Designation_FK, BaseName_FK)

WAITSTATUS (SSN_FK, Designation_FK, BaseName_FK, SequenceNumber, InFreezeZone)

MFHUNIT (UnitID, StreetNumber, ApartmentNumber, StreetName, NeighborhoodName_FK, Occupied, M-Code, IssuedToFacilitiesDate, COMCompletionDate, DateAvailable, ReturnDate, YearBuilt, FACNum, InService, Remarks, FloorPlanCode_FK, NeighborhoodName_FK)

WAITINGLIST (Designation_FK, BaseName_FK, PercentInFreezeZone)

ASSIGNMENT (UnitID_FK, SSN_FK, DateAssigned, VacatedDate, FinalInspectionDate, ClearanceInspectionDate, Pre-inspectionDate, DateRenewed, COMCost, UserID_FK)

FAMILYMEMBER (SSN_FK , FirstName, Birthdate, MI, LastName, Relation)

FLOORPLAN (FloorPlanCode, FPDescription, Designation_FK, NumberOfBedrooms, NumberOfBathrooms, Floors, SqFt, Attic, Firepl, Basemt, Carpet, Garage, Hardfl, Remarks)

APPLIANCE (ApplianceID, TypeOfAppliance, SerialNumber, Manufacturer, InstallDate, Warranty, UnitID_FK)

BASE (BaseName, Description)

ZIPCODE (Zip, City, State)

NEIGHBORHOOD (NeighborhoodName, BaseName_FK, Zip_FK)

HOUSINGCATEGORY (Designation, BaseName_FK, NumberOfBedrooms)

PRIVATERENTALPROPERTY (PropertyID, StreetNumber, ApartmentNumber, StreetName, Zip_FK, ComplexName, DateOfAvailability, Available, FloorPlanType, SqFt, Rent, PetsAllowed, NumberOfBedrooms, NumberOfBath, Attic, Firepl, Basemt, Carpet, Garage, Hardfl, Remarks, OwnerCode_FK)

OWNER (OwnerCode, PointOfContact, Street, Zip_FK, Phone, Fax, Remarks)

SYSTEMUSER (UserID, UserPassword, LastName, FirstName, MI)

HMS (UserID_FK, AccessLevel)

HFS (UserID_FK, AccessLevel)

SYSTEMADMINISTRATOR (UserID_FK, AccessLevel)

APPENDIX D. REPORTS

Actual Wait List

Category
2JEQMC

Control Date	Last Name	First	Grade	Defer Date
11/21/2000	AYALA	CARLOS	E3	
12/11/2000	BAILLIEUL	ALEXANDER	E4	
2/9/2001	WILSON	VERA	E2	

Category
2JOLM

Control Date	Last Name	First	Grade	Defer Date
6/1/2000	HUBBARD	FRANKLIN	O3	6/1/2001
12/27/2000	SAVUR	MEHMET	O2	
12/27/2000	AYVAT	BIROL	O2	
1/6/2001	GORDON	DAN	O3	

Category
2JOQMC

Control Date	Last Name	First	Grade	Defer Date
4/7/2000	KELLER	JOE	O3	4/27/2001
2/6/2001	HUNTINGTON	MIKI	O3	5/1/2001

Category
2SEQMC

Control Date	Last Name	First	Grade	Defer Date
3/20/2001	ONWUKA	CHUDY	E5	10/18/2001

Category
3FGLM

Control Date	Last Name	First	Grade	Defer Date
6/19/2000	STAGGS	CARL	O5	6/1/2001
9/15/2000	HOR	CHEONG	O4	
12/27/2000	BULGAKOV	IGOR	O4	12/30/2001
3/1/2001	BARRY	ROBERT	O4	
3/2/2001	SCHMIDT	JEFFREY	O4	5/15/2001
3/17/2001	MANSO	ROGERIO	O4	

Figure D-1.   Wait List Report

Housing Statistics

	Total Units	# Occupied	# Vacant	Occupancy %
Total	2,266	1,893	373	83.5

	Total Units	# Occupied	# Vacant	Occupancy %
POM	68	61	7	89.7
OMC	1,587	1,348	239	84.9
Total	1,655	1,409	246	85.1

	Total Units	# Occupied	# Vacant	Occupancy %
La Mesa	597	473	124	79.2
NPS	14	11	3	78.6
Total	611	484	127	79.2

Figure D-2.   Housing Statistics Report

Coast Guard Report

Paygrade	Last Name	First Name	MI	Address	Command	Date Assigned	Rank	Phone (w)
E2	NAVAL	DANIEL	J	100 OKINAWA ROAD	USCG CUTTER LONG ISLAN	5/13/2000		(831) 647-737
E3	LOPEZ	STEVE	H	194 LUZON ROAD	USCG CUTTER LONG ISLAN	8/5/1999	SN	
E3	LEWMAN	DAVID	C	132 NOUMEA ROAD	USCG STATION MONTEREY	8/22/1999	SN	(831) 647-730
E4	DILA	ERIC	A	189 NOUMEA ROAD	USCG STATION MONTEREY	11/22/1999	PO3	(831) 647-730
E4	MORINAGA	JASON	A	165 NORMANDY ROAD	USCG STATION MONTEREY	4/18/2001	MK3	(831) 647-730
E4	JONES	JON-MICHAE		165 NORMANDY ROAD	USCG CUTTER LONG ISLAN	5/13/1999	E4	(408) 325-384
E4	BRUTON	GARY	J	114 LUZON ROAD	USCG STATION MONTEREY	7/19/2000	PO3	(831) 370-218
E4	MARMONT	LINDSEY	D	180 BATAAN ROAD	USCG STATION MONTEREY	2/27/2001	PO3	
E5	WUSSEN	CHRISTOPHE	J	328 CARENTAN ROAD	USCGC LONG ISLAND	8/7/1998	BM2	( ) -831
E5	DULEMBA	JEANNIE	K	130 CORREGIDOR ROAD	USCG STATION MONTEREY	6/21/2000		(305) 292-885
E5	SPRING	DENNIS	B	199 CORREGIDOR ROAD	USCGC LONG ISLAND	7/20/1998	EM2	(831) 647-730
E5	PERIGO	MICHAEL	N	103 BOUGANVILLE ROAD	USCG STATION MONTEREY	4/11/2000	MK2	
E5	GANGSTAD	MELVIN	W	422 BRITTANY ROAD	USCG STATION MONTEREY	4/2/2001	PO2	(831) 647-731
E5	IVEY	PERCY	L	311 BRITTANY ROAD	DMDC MONTEREY	7/13/2000	YN1	( ) -831
E5	MORRIS	JOHN	T	119 CORREGIDOR ROAD	USCGC LONG ISLAND	11/17/1997	MK2	(831) 647-730
E5	MEADE	ARTHUR	E	322 KALBORN ROAD	USCG STATION MONTEREY	6/6/2000		(831) 647-730
E5	PIERRE	MICHAEL	R	401 CARENTAN ROAD	USCG STATION MONTEREY	4/28/2000	MK1	(831) 647-735
E5	GOUGE	KENNETH	C	308 MULHEIM ROAD	USCG STATION MONTEREY	7/1/1999	E5	(831) 647-730
E5	DAVIS	LUKE	A	340 CARENTAN ROAD	USCG STATION MONTEREY	7/31/2000	BU1	(831) 647-735

Figure D-3.   Coast Guard Report

133

Mayors' Report

Neighborhood   FITCH

Address	Apt.	Street	First Name	Last	Grade	# of Dependents	SVC
219		ARDENNES CIRCLE	ROLANDO	GONZALEZ	O2	2	N
227		ARDENNES CIRCLE	KATHLEEN	JUSTER	O3	1	N
244		ARDENNES CIRCLE	WARREN	OLSON	O4	3	A
250		ARDENNES CIRCLE	JORGE	FIGUEROA	O3	1	A
251		ARDENNES CIRCLE	KIP	MCCORMICK	O4	4	A
258		ARDENNES CIRCLE	ALEX	GREIG	O3	1	N
365		ARDENNES CIRCLE	ANDY	ALLEN	O3	5	A
302		HATTEN ROAD	ADONIS	MASON	O3	2	N
315		HATTEN ROAD	MARK	SARTIN	O3	5	A
208		METZ ROAD	DAVID	MATTOX	O2	2	N
224		METZ ROAD	ASIM	TOKGOZ	O2	2	X
325		METZ ROAD	JONATHAN	HOLMGREN	O3	4	N
338		METZ ROAD	STEVEN	PERCHALSKI	O3	2	N
206		REMAGEN ROAD	EDWARD	LARKIN	O2	2	A
203		ALEUTIAN ROAD	CRAIG	BURNS	E4	1	A
211		ALEUTIAN ROAD	PATRICK	DAVIS	E2	2	A
156		CORREGIDOR ROA	STEVEN	BOZEK	E5	1	N
157		CORREGIDOR ROA	KEVIN	HOWARD	E5	1	N
158		CORREGIDOR ROA	CARLOS	TYSON	E5	1	A
184		CORREGIDOR ROA	STEVEN	SASS	E6	2	A
103		SOLOMON ROAD	LISA KAY	WESTBERRY	E5	1	N
4		GILLESPIE LANE	JOSEPH	SULLIVAN	O4	3	N

Figure D-4.   Mayors' Report

Vacate Report for December 2000

Date Vacated	Address	Apt.	Street Name	Rank	First Name	Last Name	SSN	Assigned	SVC
12/28/2000	229		TUNISIA ROAD	CPT	LANCE	ENGLET	555-03-2318	12/29/1999	A
12/15/2000	318		AACHEN ROAD	CAPT	BILLY	WATTS	315-68-6611	7/21/2000	A
12/26/2000	306		AACHEN ROAD	CPT	SCOTT	WESTERF	263-91-1617	7/9/1999	M
12/18/2000	209		ARDENNES CIRCLE	CPT	DONALD	SCHURR	164-62-3015	10/27/1999	A
12/14/2000	329		METZ ROAD	CPT	JOHN	SMLS	266-86-9561	5/6/2000	A
12/29/2000	250		ARDENNES CIRCLE	LT	DAVID	ABERNAT	261-65-5380	1/9/1998	N
12/21/2000	375		ARDENNES CIRCLE	CPT	ERIC	BLEAKIE	055-68-7527	12/7/1999	A
12/15/2000	265		ARDENNES CIRCLE	MAJ	MICHAEL	AHERN	271-76-1436	8/15/1999	M
12/7/2000	218		ARDENNES CIRCLE	MAJ	ROBERT	BUSHTA	519-78-6967	9/14/1999	M
12/20/2000	253		ARDENNES CIRCLE	MAJ	MICHAEL	KELLY	555-49-3560	5/27/1999	M
12/15/2000	249		TUNISIA ROAD	CAPT	MATHEW	VALIQUET	353-72-5772	6/8/1999	M
12/14/2000	302		HATTEN ROAD	CPT	SIMON	GARDNER	012-60-8813	6/3/1999	A
12/14/2000	211		TUNISIA ROAD	CPT	JEFFREY	MEGARG	153-68-6779	5/25/1999	M
12/14/2000	234		METZ ROAD	LT	TRACY	WHITELE	451-49-3032	7/28/1998	N
12/14/2000	275		ARDENNES CIRCLE	MAJ	CLAYTON	FISHER	335-68-9060	4/18/2000	M
12/15/2000	243		METZ ROAD	LT	PHILIP	MALONE	111-48-9126	10/14/1998	N
12/15/2000	317		HATTEN ROAD	MAJ	GREGOR	JAMES	540-86-6354	7/10/1999	A
12/13/2000	319		ARDENNES CIRCLE	LT	RUSSELL	HISER	042-48-0905	12/9/1999	N

Figure D-5.   Vacate Report

135

Occupant Report

Street #	Street	Apt.	Unit #	Last Name	First	Paygrade	Dependents
300	AACHEN ROAD		3289	WILLIAMS	CLIFTON	O3	4
301	AACHEN ROAD		3291	ANDERSON	TIM	O3	3
302	AACHEN ROAD		3292	TAYLOR	TARL	O4	5
303	AACHEN ROAD		3293	VEGTER	HENRY	O2	3
304	AACHEN ROAD		3294	GORAK	MARK	O3	4
305	AACHEN ROAD		3295	RULOF	ROBERT	O2	2
306	AACHEN ROAD		3297	DOLLOFF	KATHERINE	O3	2
307	AACHEN ROAD		3298	DEVINE	CRAIG	O3	5
308	AACHEN ROAD		3299	KUBISTA	THEODORE	O3	2
309	AACHEN ROAD		3300	DUGONE	THEODORE	O3	4
310	AACHEN ROAD		3301	DIMMICK	MATTHEW	O3	3
311	AACHEN ROAD		3302	SPENCE	EDWAR	O3	3
312	AACHEN ROAD		3304	JOHNSON	STEVEN	O3	4
313	AACHEN ROAD		3306	KENNEDY	RICHARD	O2	2
314	AACHEN ROAD		3307	MCLEAN	CHARLES	O3	2
315	AACHEN ROAD		3310	DAVIS	CHRISTOP	O3	4
316	AACHEN ROAD		3312	HARRIS	BENJAMIN	O3	3
318	AACHEN ROAD		3314	COOMBS	JOHN	O3	4
201	ALGERIA ROAD		2619	MUHAMMAD	ANGELA	O3	3
203	ALGERIA ROAD		2621	PARSONS	DARREN	O3	3
204	ALGERIA ROAD		2622	SORENSON	ANDREW	O3	3
205	ALGERIA ROAD		2623	CHI-HOON	KIM	O3	2
207	ALGERIA ROAD		2624	PATERSON	RYAN	O3	1
210	ALGERIA ROAD		2626	CELIK	GUNER	O2	1
200	ARDENNES CIRCLE		2627	SCHEIN	ARI	O3	1
201	ARDENNES CIRCLE		2628	MING	MEL	CW3	2
203	ARDENNES CIRCLE		2629	ROSS	JAMES	O3	3

Figure D-6.   Occupant Report

Non-eligible Report

Last Name	First	MI	Paygrade	SVC	PRD	Address
CAN	YUKSEL		O2	X	3/30/2000	1277 SPRUANCE ROAD
KALLOCH	AARON	E	O3	A	6/30/2000	1051 HALSEY DRIVE
KALLOCH	AARON	E	O3	A	6/30/2000	381 B BERGIN DRIVE
MEWES	TIMOTHY	J	O3	F	5/21/2001	311 HATTEN ROAD
CARRIER	KEVIN	J	O4	N	3/30/2001	12 MERVINE STREET
DONAHUE	MICHAEL	C	O3	A	5/24/2001	202 BONN ROAD
MASON	FREDERICK	J	O5	A	1/30/2000	434 LEWIS ROAD
CONFER	JASON	R	E6	M	5/20/2001	401 RADDEN ROAD
ELEAZER	PATRICIA	A	E5	F	6/25/2000	306 BRITTANY ROAD
MURPHY	DANIEL	E	O3	N	5/26/2000	382 D BERGIN DRIVE
ROOKE	JEFFREY	T	E3	M	5/17/2001	198 NOUMEA ROAD
MAGUIRE	GREG	M	O4	N	6/30/2000	233 ANZIO ROAD
BYERS	MICHAEL	R	E5	N	5/1/2001	102 SOLOMON ROAD
LANNAMANN	DANIEL	L	O3	N	5/23/2001	584 C MICHELSON RO
KORENOSKI	JAMES	J	CW1	A	5/17/2001	300 ARLONCOURT RO
MALLOY	ARREN	L	E6	M	4/12/2001	310 KALBORN ROAD
RAUSCH	JEFFERY	S	E5	A	4/30/2001	416 KALBORN ROAD
GONZALEZ	FRANK	R	E5	F	5/20/2001	125 CORREGIDOR ROA

Monday, May 28, 2001

Figure D-7.   Non-eligible Report

137

Utilization Report

	# Occupied	# Hold	# Available	Utilization Rate
June 2000	1922	75	229	86.3%
July 2000	1901	73	252	85.4%
August 2000	1927	65	234	86.6%
September 2000	1929	65	232	86.7%
October 2000	1917	68	241	86.1%
November 2000	1913	67	246	85.9%
December 2000	1910	65	250	85.8%
January 2001	1915	80	231	86.0%
February 2001	1925	75	226	86.5%
March 2001	1928	71	227	86.6%
April 2001	1902	70	254	85.4%
May 2001	1893	68	265	85.0%

Figure D-8.   Utilization Report

School Children (by Base)

Base LM

First Name	Last	Relation	Birth Date	Age
RYLAND	DARLING	S	5/29/1997	4
PHILLIP	PEITSO	S	5/19/1997	4
KIMBERLY	RUEDA	D	5/15/1997	4
HALEY	WOFFORD	D	5/1/1997	4
ALEXANDRA	BERRY	D	4/27/1997	4
CONOR	ARMSTRONG	S	4/7/1997	4
ALLISON	NASH	D	4/4/1997	4
MEGHAN	HOGAN	D	3/17/1997	4
NICOLAOS	PAVLOUDIS	S	3/16/1997	4
MARIAM	ALQETAMI	D	3/13/1997	4
NICHOLAS	PONCE	S	3/5/1997	4
VICTORIA	MORRIS	D	2/28/1997	4
emily	branstetter	D	2/17/1997	4
ASHLEY	PAYNE	D	2/14/1997	4
NATHAN	PECK	S	2/10/1997	4
JYLE	SANDERS	S	2/8/1997	4
EMILY	PETERS	D	2/8/1997	4
SAMANTHA	PRICE	D	2/8/1997	4
NICOLAS	GRANDJEAN	S	1/21/1997	4
KATRINA	MULLER	D	1/15/1997	4
DILLON	O'BRIEN	S	1/9/1997	4
MIA	STREZZI	D	1/6/1997	4
JAIME	FOSTER	D	1/3/1997	4
TIMOTHY	DEVLIN	S	1/3/1997	4
CHIEN	YEN	S	1/3/1997	4

Monday, May 28, 2001

Figure D-9.   School Children Report

139

Assignment History

Base   LM

Unit # 1   370   A   BERGIN DRIVE

Date Assigned	Date Vacated	SSN	Last Name
6/6/1994	4/25/1996	474-82-6238	OCASIO
6/28/1996	9/17/1998	522-19-7641	WAKEFIELD
10/21/1998	6/28/2000	478-74-1048	MCCABE

Unit # 2   370   B   BERGIN DRIVE

Date Assigned	Date Vacated	SSN	Last Name
1/10/1994	3/27/1996	230-04-6746	PROTACIO
6/20/1996	1/14/1998	226-98-9801	WHITESCARVER
3/3/1998	6/30/1999	397-74-7115	ZANDER
7/6/1999	12/21/2000	218-82-0518	ZAWISLAK

Unit # 3   370   C   BERGIN DRIVE

Date Assigned	Date Vacated	SSN	Last Name
12/22/1993	8/6/1995	187-48-1473	MARINO
1/24/1996	4/2/1998	557-45-4396	JERAULD
5/23/1998	8/4/2000	535-90-3379	SHEA

Unit # 4   370   D   BERGIN DRIVE

Date Assigned	Date Vacated	SSN	Last Name
8/7/1995	7/30/1996	219-70-6830	JONES
8/29/1996	3/25/1999	472-74-6815	DELGADO
6/1/1999		485-92-1533	PECK

Unit # 5   370   E   BERGIN DRIVE

Date Assigned	Date Vacated	SSN	Last Name
6/30/1994	10/2/1996	195-62-1853	FREE
5/9/1995	4/3/1996	083-60-7612	WEST

Sunday, May 27, 2001

Figure D-10.   Assignment History Report

140

LIST OF REFERENCES

Ambler, Scott W. and Constantine, Larry L., *The Unified Process Inception Phase: Best Practices in Implementing the UP*, CMP Media, Inc., 2000.

Aronson, Jay E. and Turban, E., *Decision Support Systems and Intelligent Systems*, Prentice Hall, 1998.

Bruegge, Bernd and Dutoit, Allen H, *Object-Oriented Software Engineering: Conquering Complex and Changing Systems*, Prentice Hall, 2000.

Fowler, Martin and Scott, Kendall, *UML Distilled: Applying the Standard Object Modeling Language*, Addison Wesley Longman, Inc., 1997

Kroenke, David M., *Database Processing: Fundamentals, Design, and Implementation*, Prentice Hall, 2000.

Logistics Management Institute Report NA003L1, *Technical Point Paper: Analysis of Electronic Family Housing and Archibus*, by John Harris, January 2001.

Telephone conversation between Leidle, Lael D., Computer Specialist, Pacific Division West Coast Housing Division, Naval Facilities Engineering Command, Housing Division, and the author, 18 January 2001.

"What is GIS?" Environmental Systems Research Institute (ESRI) Inc. Web Site, [www.gis.com/whatisgis.index.html]. January 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
    8725 John J. Kingman Road, Suite 0944
    Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
    Naval Postgraduate School
    411 Dyer Road
    Monterey, CA 93943-5101

3. CPT Jonalan Brickey 2
    2510 Kenwood Boulevard
    Toledo, OH 43606

4. Professor Daniel R. Dolk, Code IS/Dk 1
    Naval Postgraduate School
    Monterey, CA 93943-5118

5. MAJ James E. Illingworth 2
    U.S. Army TRADOC Analysis Center
    Monterey, CA 93943

6. NSAMB Housing Office 1
    Attn: Brad Collier
    1357 Leahy Road
    Monterey, CA 93940-4882

7. Chair, IS Academic Group, Code IS 1
    Naval Postgraduate School
    Monterey, CA 93943-5118